Programming without coding technology

تكنولوجيا البرمجة بدون كود

(1) Mahmoud Programming Language

(2) RPWI Environment

(3) DoubleS (Super Server) Paradigm

(١) لغة البرمجة محمود

(٢) بيئة البرمجة بدون كود

(٣) نمط البرمجة الخادم الممتاز



Version 1.0
(Stable) Rev. 8

الاصدارة الاولى
المراجعة الثامنة

http://www.sourceforge.net/projects/doublesvsoop

By
Mahmoud Fayed
msfclipper@users.sourceforge.net

جدول المحتويات

INTRODUCTION مقدمة

Programming is giving instructions to computer to perform the required job, the process of programming have more than one stage from analysis and design to implementation and test.

البرمجة هى اعطاء التعليمات للحاسب لكى ينفذ العملية المطلوبة وعملية البرمجة (انشاء التطبيقات) تتطلب اكثر من مرحلة مرورا بالتحليل والتصميم الى التطبيق والاختبار

The most common way to do programming is (writing code) which our compiler under usage can understand based on the syntax of the programming language.

الطريقة الشائعة للبرمجة (التطبيقية) هى كتابة الكود الذى يفهمه المترجم من خلال القواعد النحوية للغة البرمجة

To write code you need software called a Code Editor (CE), which Enable the programmer to organize his/her source code file, Modern Code Editors comes with nice features like (Auto completion, Syntax highlighting & IntelliSense) which enable the programmer to work more easier and faster.

لكتابة الكود ان بحاجة الى محرر للكود الذى يمكنك كمبرمج من تنظيم ملف الشفيرة المصدرية الخاص بك ومحررات الكود الحديثة تاتى بملامح رائعة مثل الاكمال التلقائى واظهار القواعد النحوية والاحساس الادراكى الذى يمكن المبرمج من العمل بسهولة وسرعة

Since coding requires from the programmer the attention and effort, a lot of researchers focus on this point, some researches work on making coding less harmful while other researchers try to eliminate and reduce the necessary and the need of coding which we say about (Programming without coding)

لان عملية كتابة الكود تتطلب من المبرمج التركيز والانتباه فان هناك العديد من الابحاث تركز على تلك النقطة واتجه بعض الباحثين الى تسهيل الكود بينما اتجه اخرون الى الابتعاد عن الكود مما يسمى بالبرمجة بدون كود

There are a lot of steps towards (Programming without coding), the first step was (Copy & Paste) operations where the programmer write the code one time then copy it and use it another time in another program, and this without doubt is very bad solution and seems to be the heart of problems.

هناك العديد من الخطوات نحو البرمجة بدون كود – كانت الخطوة الاولى هى عمليات النسخ واللصق لجزء من الكود وبالتاكيد هذه الخطوة سببا للعديد من المشاكل.

The second step was the (Code Bank) where the programmer store his common code in one place to get it quickly and still do (Copy & Paste) Operations.

الخطوة الثانية هى عبار عن بنك الكود حيث يخزن المبرمج الشفيرة الخاصة به فى مكان واحد يحصل منه بسرعة على الكود ولكن مازال يقوم بعمليات النسخ واللصق

The third step was the (Library) where a one or more of functions are encapsulated together and are available for use directly by the programmer after referring to the

`

library header file in the source code and referring to the path of the library file for usage by the linker to produce executable file.

الخطوة الثالثة هى المكتبة حيث يتم تجميع الدوال معا ويستخدمها المبرمج بعد الاشارة الى ملفات تعريف المكتبة بالاضافة الى اسم المكتبة اثناء عملية الربط مع نظام التشغيل

The forth step was the (Template) which is a mask of code, which can be used more than one time.

الخطوة الرابعة هى القالب الذى هو عبارة عن قناع للكود يمكن استخدام اكثر من مرة.

The fifth step was the (Code Generator) which uses (Templates) and get data from the programmer to generate the code automatically.

الخطوة الخامسة هى مولد الكود الذى يعتمد على القالب وعلى استقبال بيانات من المبرمج لتوليد الكود تلقائيا

The Sixth step was the (Designers) like (Database designer, Class Designer, Form Designer, Menu Designer, Report Designer,…etc)

الخطوة السادسة هى المصممات مثل مصمم قواعد البيانات والفصائل والنماذج والقوائم والتقارير.

The seventh step was the (Wizard) which is a group of interaction pages that end up with generating designer files and source code.

الخطوة السابعة هى المعالج الذى يعتمد على مجموعة من الصفحات التفاعلية التى تنتهى بتوليد ملفات تصميم وملفات شفيرة مصدرية

In these days the (IDE and Frameworks) are used to integrate Libraries, Templates, Code Generators, Wizards and Designers together.

But still the Programmer/Developer needs to write code to complete his task (the required job).

فى هذه الايام انتشر استخدام بيئة التطوير المتكاملة مع هياكل العمل حيث يتم عمل تكامل بين المكتبات والقوالب ومولدات الاكواد والمعالجات والمصممات.
لكن مازال المبرمج بحاجة الى كتابة الكود لكى ينهى مهمته المطلوبة

Programming without coding technology presents the first general purpose programming without coding language (Mahmoud), programming without coding environment (RPWI) and programming paradigm (DoubleS = Super Server) for developing software applications 100% without coding by providing full isolation from the programming language syntax where the programming effort done through interaction instead of writing code and (You need to know procedure instead of being declarative).

تكنولوجيا البرمجة بدون كود تقدم اول لغة برمجة عامة بدون كود (محمود) وبيئة البرمجة بدون كود و نمط البرمجة الخادم الممتاز لكى يتم انتاج التطبيقات 100% بدون كود مع تقديم العزل الكامل من قواعد لغة البرمجة حيث ان مجهود البرمجة يتم من خلال التفاعل مع الواجهة الرسومية بدلا من كتابة الكود اى انك بحاجة الى معرفة خطوات روتينية بدلا من الاعلان عن ماتريد.

`

Since Programming without coding technology change the programming world from it's heart (The Code), it's expected form this new technology to present something different replace what we use while coding.

لان تكنولوجيا البرمجة بدون كود تغير عالم البرمجة من قلبه (الكود) فانه من المتوقع ان تقدم شيئا بديلا يحل محل ماهو موجود اثناء كتابة الكود

Coding البرمجة بالكود	**Programming without coding البرمجة بدون كود**
Job/Problem المشكلة او الوظيفة	Goal الهدف
Lines of code + Comments عددا من الاسطر فى الكود	Step خطوة
Code Editor محرر الكود	Goal Designer مصمم الهدف
Writing Code كتابة الكود	Interaction التفاعل
Compiler المترجم	Interactive Language (GUI) لغة تفاعلية تتمثل فى واجهة رسومية
Programming Language Syntax قواعد لغة البرمجة	You need to know procedure انت بحاجة الى معرفة الاجراءات الروتينية
Wizards & Code Generators المعالجات ومولدات الاكواد	Transporter الناقل

Table 1: Coding vs. programming without coding

Goal Designer vs. Code Editor
مصمم الهدف بديلا لمحرر الكود

Goal Designer is the tool of programming without coding
Inside programming without coding technology

مصمم الهدف هو اداة البرمجة بدون كود داخل تكنولوجيا البرمجة بدون كود

- Goal Designer is Full Replacement for Code Editor بديل كامل لمحرر الكود
- Goal Designer can work side by side with Code Editor if this feature is required يمكن ان يعمل بجانب محرر الكود اذا كانت تلك الخاصية مطلوبة
- Goal Designer manage your source code file in Modern way, each goal is group of steps, while each step contains its block of code يدير الشفيرة المصدرية الخاصة بك بطريقة حديثة حيث ان الهدف مجموعة من الخطوات وكل خطوة عبارة عن مجموعة من الاكواد
- Using code extractor application, you can create source code file by extracting code from one goal or more. باستخدام مستخرج الكود يمكنك ان تستخرج الكود من هدف او مجموعة من الاهداف

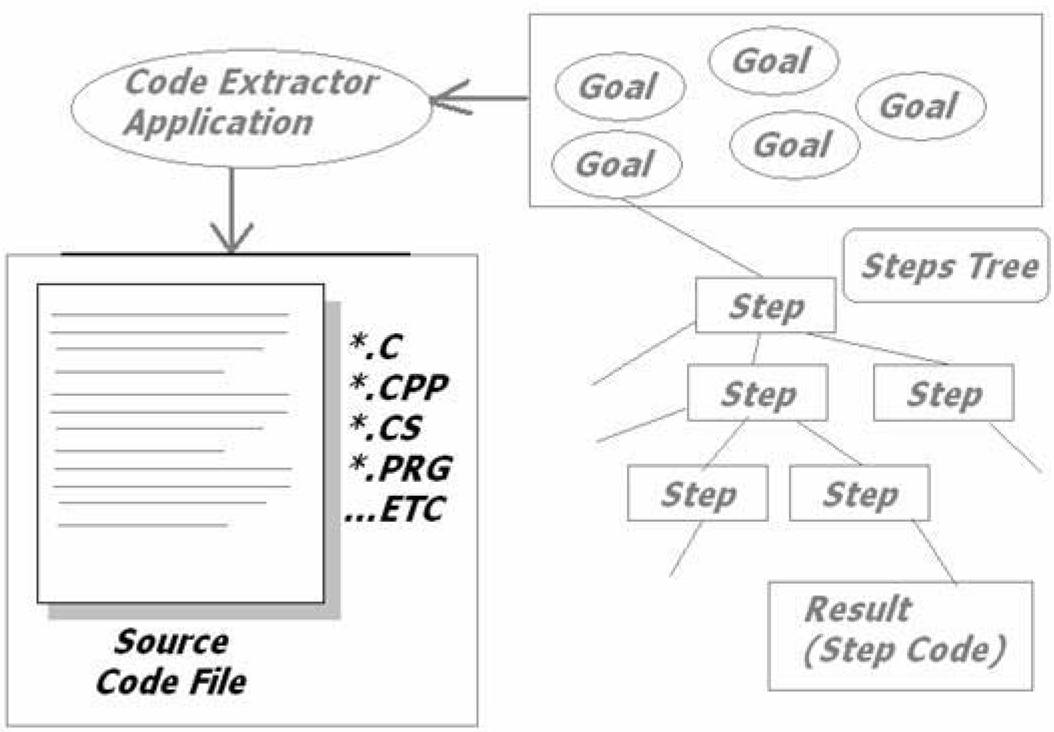


Fig. 1 (The idea of Goal Designer)

Goal Designer Features ملامح مصمم الاهداف:-

- Moving between different Goals التنقل بين الاهداف المختلفة
- Create New Step انشاء خطوة جديدة

- Edit Step name تحرير اسم الخطوة
- Delete Step حذف الخطوة
- Move Step Up in the steps tree نقل الخطوة لاعلى
- Move Step Down in the steps tree نقل الخطوة لاسفل
- Interaction to determine the job of the step التفاعل لتحديد وظيفة الخطوة
- Step Details (Data, Analysis, Information, Code, Info) لكل خطوة تفاصيل تحددها
- Copy Step or group of steps into buffer نسخ مجموعة من الخطوات
- Insert Step or group of steps from buffer to the steps tree لصق الخطوات

Interaction vs. writing code
التفاعل بدلا من كتابة الكود

Interaction used to determine the job of the step التفاعل يستخدم لتحديد وظيفة الخطوة
The procedure of the interaction process عملية التفاعل تتم كالتالى

- Determine step تحديد الخطوة
- Determine component to interact with تحديد المكون الذى نتفاعل معه
- Enter data required by the selected component ادخال البيانات المطلوبة

As a result of the interaction process نتيجة عملية التفاعل

- The code will generated automatically in the (background) توليد الكود فى الخلفية
- Information will generated automatically so the programmer can understand the flow of the program without the need to look at the code توليد المعلومات التى تمثل الكود وتجعلنا قادرين على متابعة سير البرنامج بدون الحاجة الى النظر الى الكود

The question now, what is new over the legacy code generator and wizards?
السؤال الان ماهو الجديد عن مولدات الكود والمعاجات

- You don't need to look at code at all لست بحاجة للنظر الى الكود على الاطلاق
- You can modify your work without looking at code at all يمكنك تعديل عملك بدون الحاجة للتعامل مع الكود
- You can control the flow of the program where you can (Move steps up, move steps down, Copy & Paste steps without the need to look at the code at all) يمكنك التحكم فى سير البرنامج واجراء عمليات النسخ واللصق بدون الحاجة للنظر الى الكود

Programming without coding technique
اسلوب البرمجة بدون كود

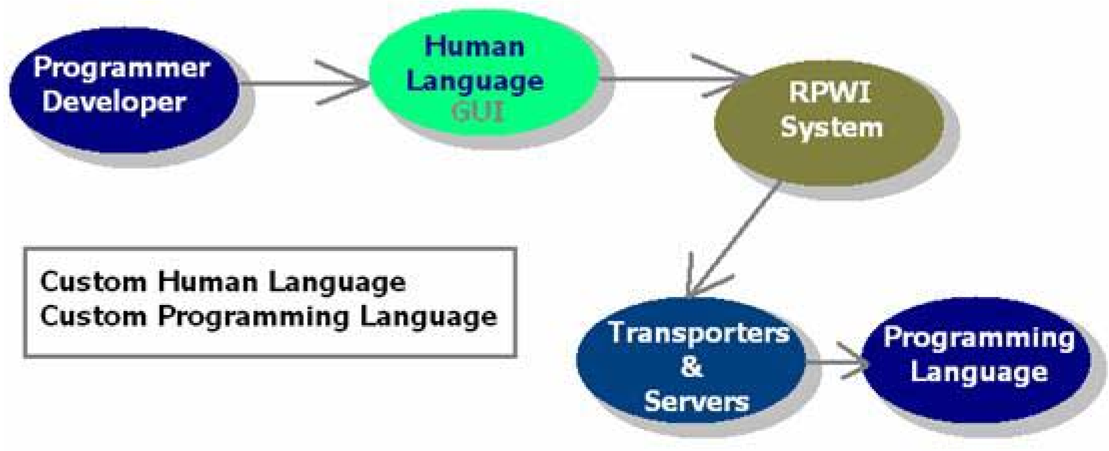


Fig. 2 (Programming without coding technique)

- Isolation from programming language syntax عزل المبرمج عن قواعد اللغة
- Custom User Interface (Human Language) واجهة مستخدم خاصة بلغة انسانية كالمعتاد
- Based on Programming Language Behind the scene الاعتماد على لغة برمجة فى الخلفية
- Use Any Programming Language Behind the scene يمكن استخدام اى لغة فى الخلفية
- No Restrictions & Without Limitations لا حدود ولا قيود

The Transporter
الناقل

In Programming world, there is Low Level Programming Languages and High Level Programming Languages, Also in programming without coding technology which presents new Generation of programming languages (Interactive Languages) there is Low Level RPWI and High Level RPWI Languages

فى عالم البرمجة هناك لغة منخفضة المستوى وعالية المستوى وهذه مقارنة نسبية بين اللغات وكذلك فى تكنولوجيا البرمجة بدون كود التى تقدم جيل جديد من لغات البرمجة وهو اللغات التفاعلية هناك لغات برمجة بدون كود منخفضة المستوى واخرى عالية المستوى

RPWI means " (R)eal (P)rogramming (W)ithout cod(I)ng ", Where you don't need to write code, or to study programming language syntax because you develop your software in RPWI Environment through interaction, i.e. GUI instead of writing Code.

Low Level RPWI requires from the developer/programmer to know programming concepts, and requires that you understand the features and rules presented by the programming language under usage because the interactive language just present GUI

`

(in human language) to all features available from the programming language under usage.

اللغات التفاعلية المنخفضة المستوى فى البرمجة بدون كود تعتمد على فهم مستخدمها لتفاصيل فنية عن لغة البرمجة المستخدمة فى الخلفية

Also all Components (Servers/Transporters) available in interactive language which is Low Level RPWI, will be like direct interface to features available by the programming language, so the code generated from any component (Server/Transporter) will do a job that may not make sense in the general application, but only the developer/programmer who uses the component can understand the generated code.

High Level RPWI doesn't require from the developer/programmer to know programming concepts, no need to know the rules of the programming language under usage, because here the interactive language that is High Level RPWI, provides GUI (in human language) + components (Servers/Transporters) which generates code which do expected job anyone can fell with it and make a sense in the general application under development.

اللغات التفاعلية عالية المستوى فى البرمجة بدون كود تخاطب المستخدم ولاتتطلب فهم لتفاصيل فنية كثيرة.

Transporter, this name because it's used for transporting the programming language to RPWI Environment.

Transporter differs from the wizard and code generators, its upgrade to them because transporters work together in solving the same problem which maximize the power and free the space to an infinite number of applications that we can develop which provide us with unlimited and general purpose programming environment.

Programming without coding technology architecture
معمارية تكنولوجيا البرمجة بدون كود

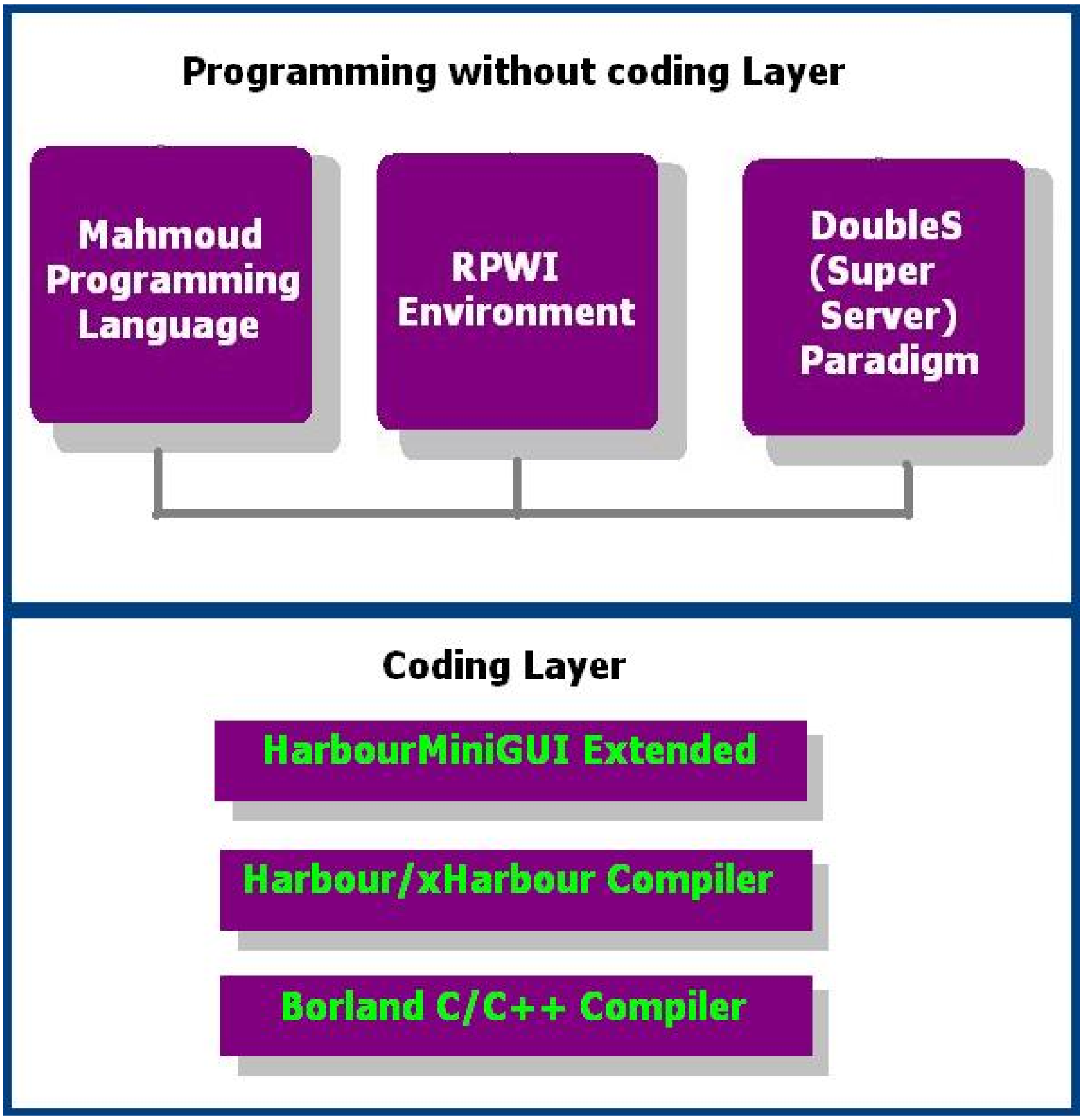


Fig.3 (Programming without coding technology architecture)

(1) Programming without coding layer طبقة البرمجة بدون كود
   a. Mahmoud Programming Language
   b. RPWI Environment
   c. DoubleS (Super Server) Paradigm

(2) Coding Layer طبقة البرمجة بالكود
   a. HarbourMiniGUI Extended (GUI Framework)
   b. Harbour/xHarbour Compiler
   c. Borland C/C++ Compiler

`

How to get Programming without coding technology?
كيفية الحصول على تكنولوجيا البرمجة بدون كود

You can download programming without coding technology from
URL: http://www.sourceforge.net/projects/doublesvsoop
It's Free and Open Source.

انها مجانية ومفتوحة المصدر يمكنك الخصول عليها من الموقع الخاص بها والمذكور اعلاه

The tools used for building the technology
الادوات المستخدمة فى بناء تكنولوجيا البرمجة بدون كود

1 – Microsoft Visual FoxPro 9.0 SP2
2 – HarbourMiniGUI Extended (GUI Framework)
3 – Harbour/xHarbour Compiler
4 – Borland C/C++

1 Mahmoud Programming Language

Introduction مقدمة

Mahmoud programming language is the first general purpose interactive (Based on interaction through GUI instead of writing code) programming Language developed and hosted in the RPWI Environment.

لغة البرمجة محمود هى اول لغة برمجة عامة تفاعلية تم تطويرها واستضافتها داخل بيئة البرمجة بدون كود

This language is a good example about what we can do inside RPWI Environment, where we can develop and use interactive languages in a Very easy manner, so you can develop your own programming language That carry your name (or any other name you like), and you can hack Mahmoud programming language and start doing whatever change you like to do.

اللغة عبارة عن مثال لما يمكن عمله داخل بيئة البرمجة بدون كود حيث يمكن تطوير واستخدام اللغات التفاعلية بطريقة بسيطة جدا ولذلك يمكنك تطوير لغة البرمجة التفاعلية الخاصة بك التى تحمل اسمك او اى اسم اخر تجده مناسبا ويمكنك اختراق لغة البرمجة محمود والبداية بعمل التعديلات التى تريدها.

The Language provides us with great power from the ability to develop console applications to GUI applications and from simple mathematics applications to complex applications including database applications from desktop to the network.

The language power are not limited or restricted, it's open architecture where you can extended the language by adding new components using RPWI Environment designers and/or you can use extension components that enable you to call DLL functions, use COM objects and/or do OLE Automation.

Now we are going to learn how to use this Modern programming language to explore its features and simply present its power.

اللغة تقدم العديد من الملامح الجيدة التى تجعلها لغة بسيطة وقوية فى نفس الوقت.

`

Hello World:-

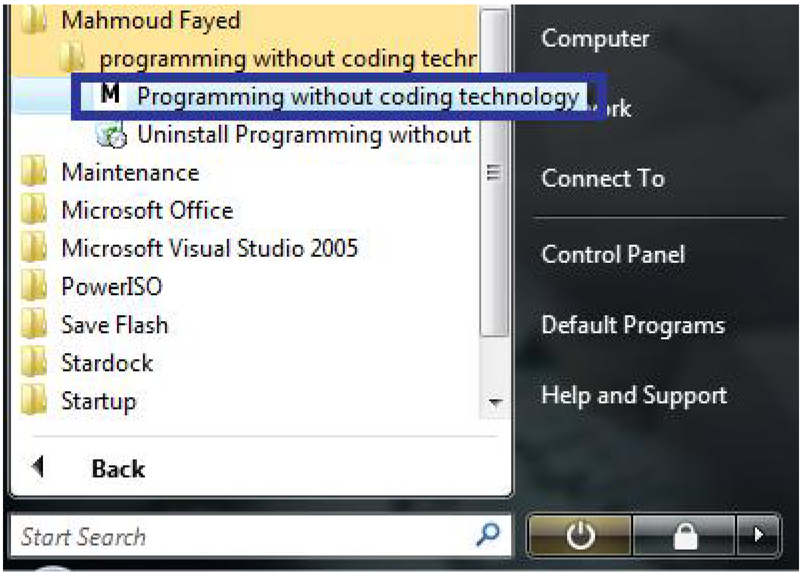
(1) Procedure
(2) Screen Shots

(1) Procedure:-

- Run Programming without coding technology software
- Wait to see the logo screen
- Click “ok” to close the about window
- Click New
- Select “Console Application – without coding”
- Click “New”
- Determine the file name (*.SSF)
- Select Goal Designer
- Click “Interact” button
- Select Domain (Console Applications)
- Select Component (Basic Input/Output)
- Click ok
- Check (Print text on new line)
- Write the text for displaying on the screen
- Click “ok” button
- Select the first step
- Click “Interact” button
- Select Domain (Console Applications)
- Select Component (Wait Key/Seconds)
- Click “ok” button
- Check “wait n seconds”
- Determine the number of seconds to wait
- Click ok
- Run the program
- Wait for saving & building the application
- Have fun Seeing the application in the runtime

This long procedure requires less than 30 seconds and it’s very easy to learn and remember, is it?

(2) Screen shots:-



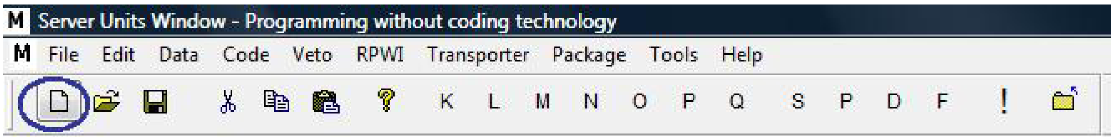
Run Programming without coding technology software



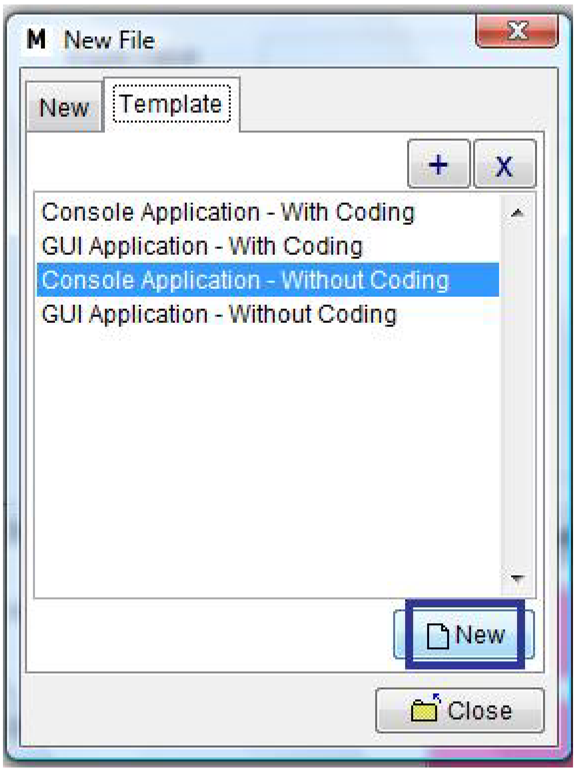
Wait to see the logo screen



Click "ok" to close the about window



Click New



Select "Console Application – without coding" then Click "New"

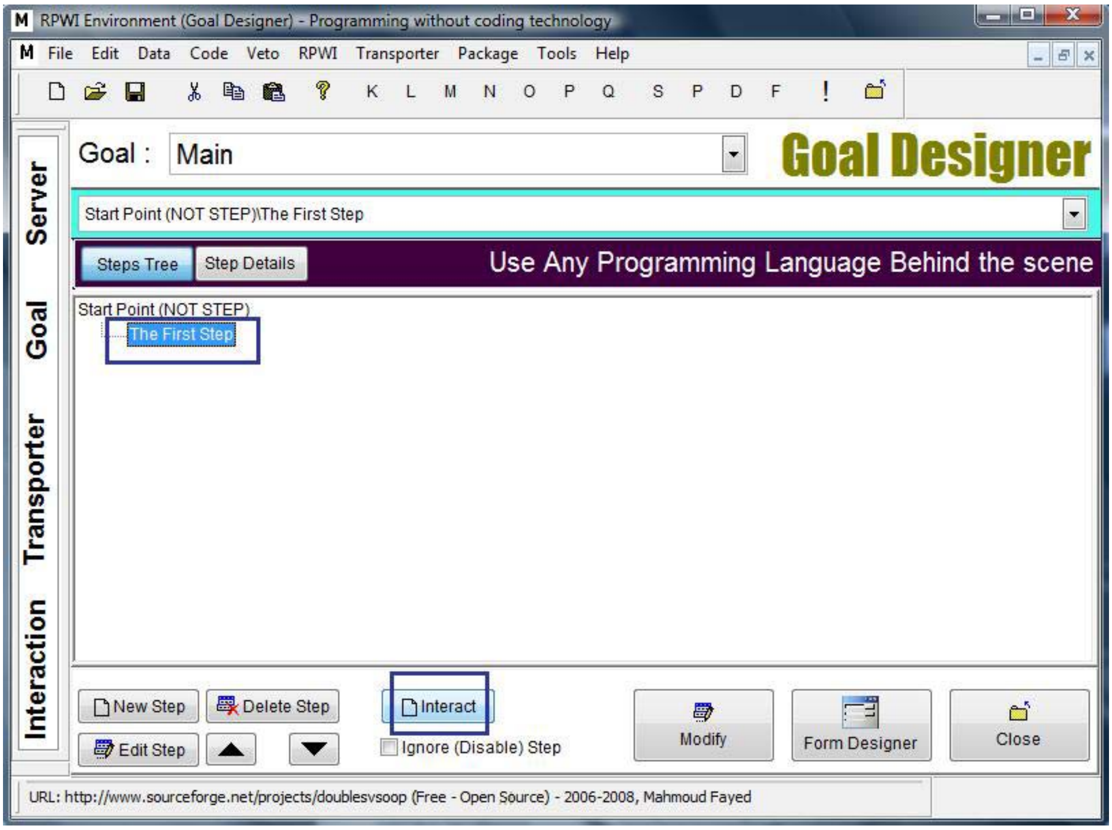
`



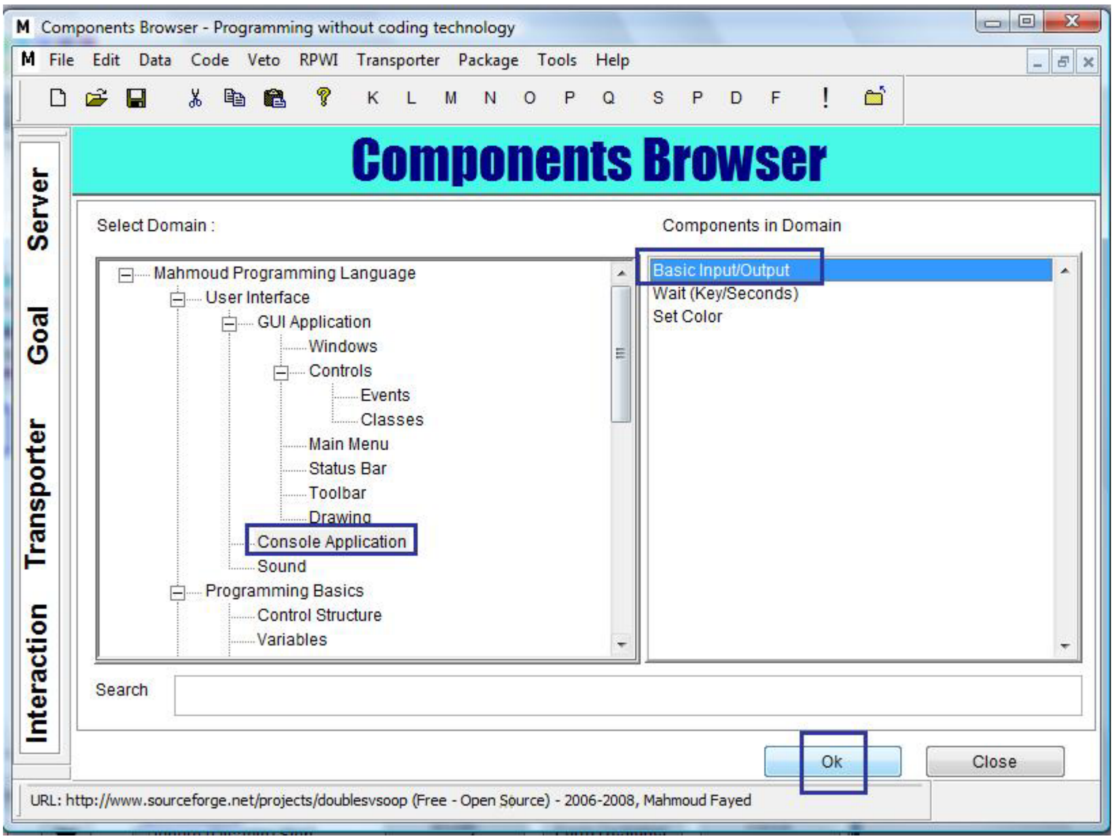
Determine the file name (*.SSF)



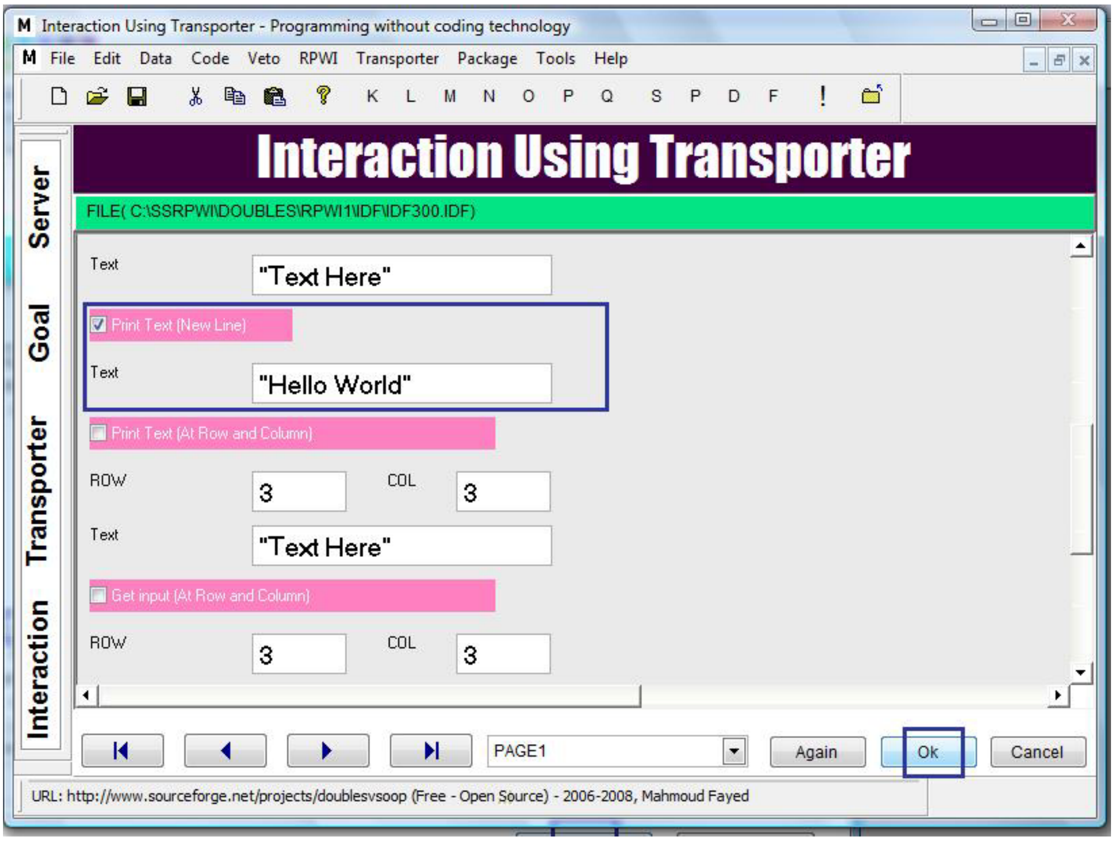
Select Goal Designer



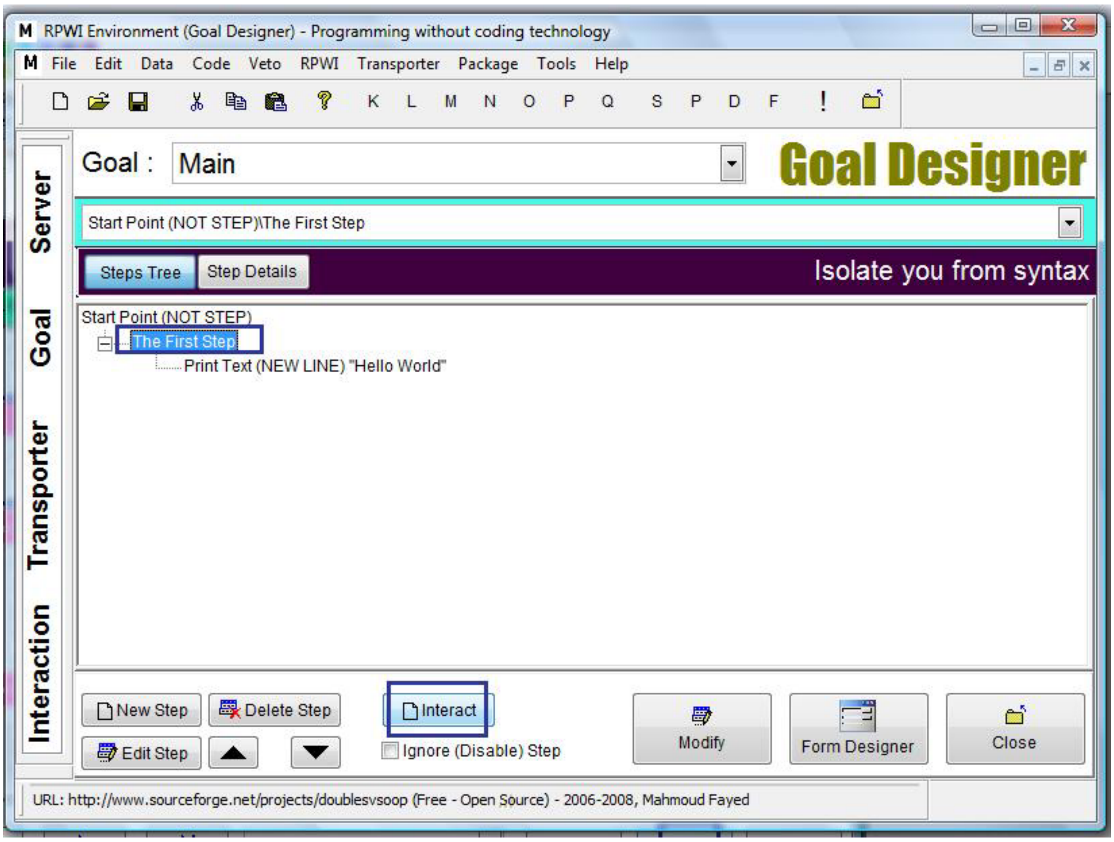
Click "Interact" button



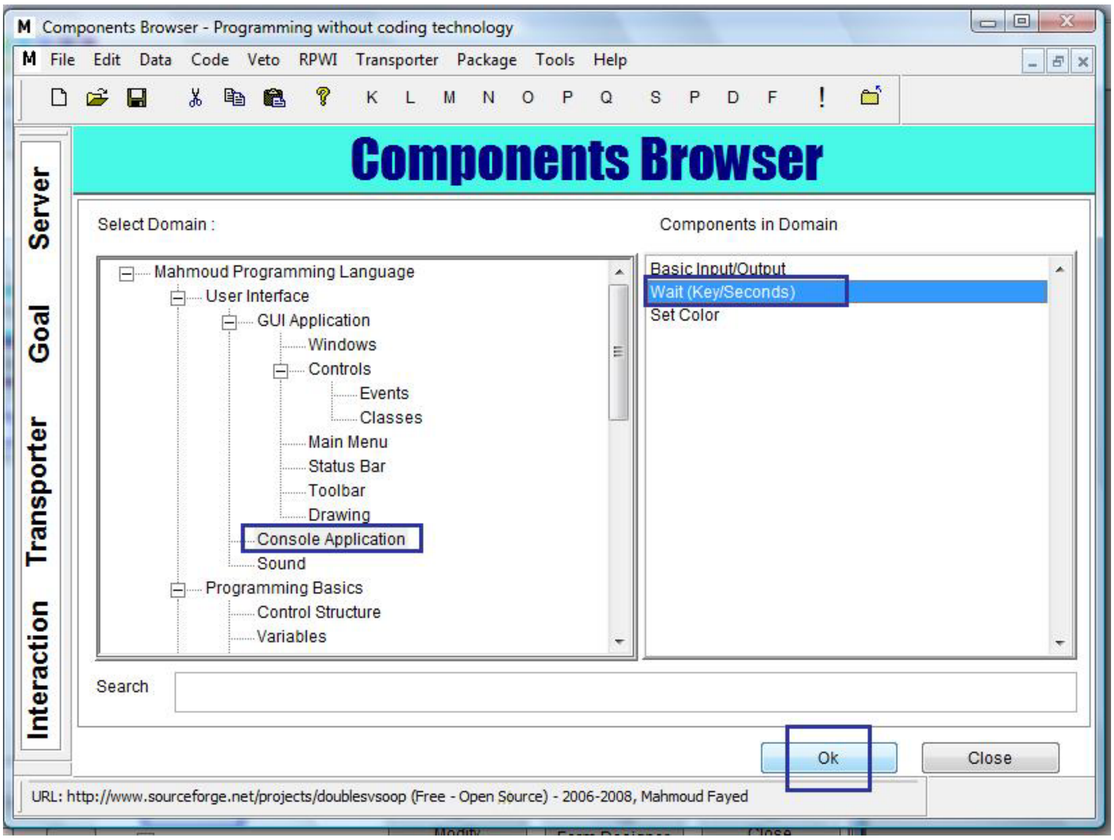
Select Domain (Console Applications) then Select Component (Basic Input/Output) then Click ok



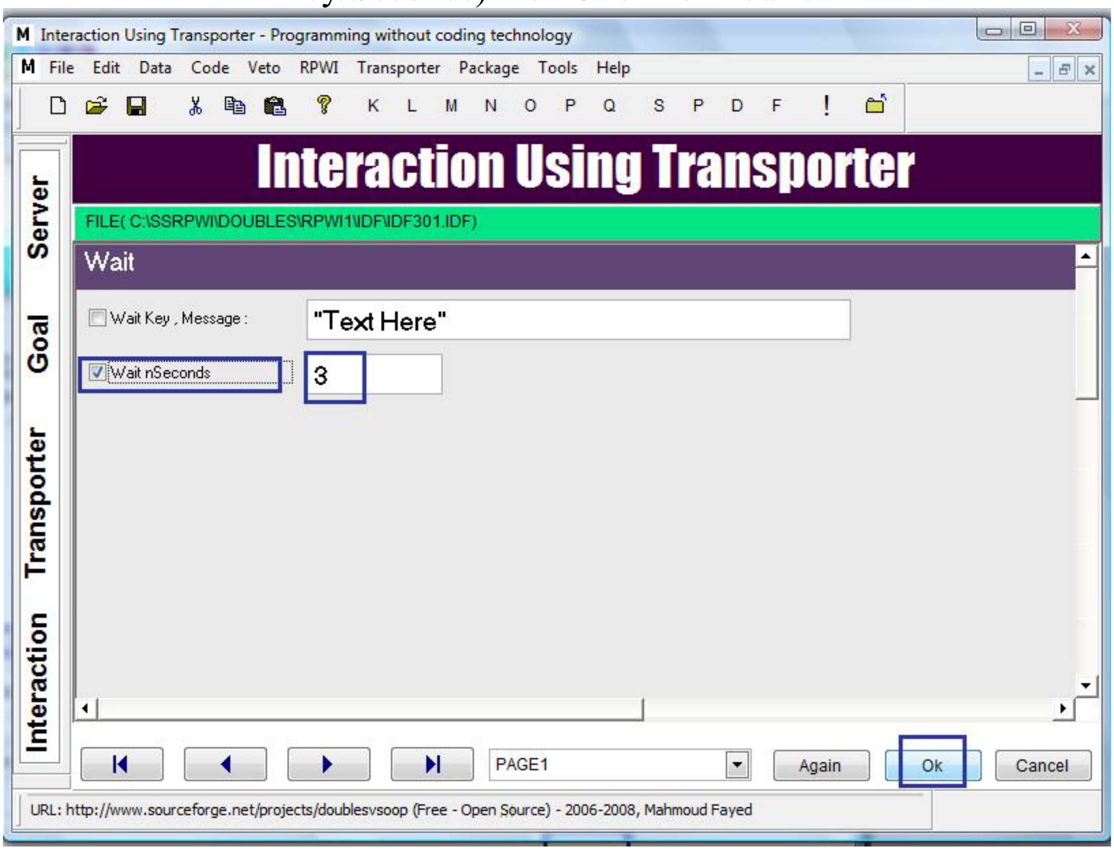
Check (Print text on new line) then Write the text for displaying on the screen then Click "ok" button



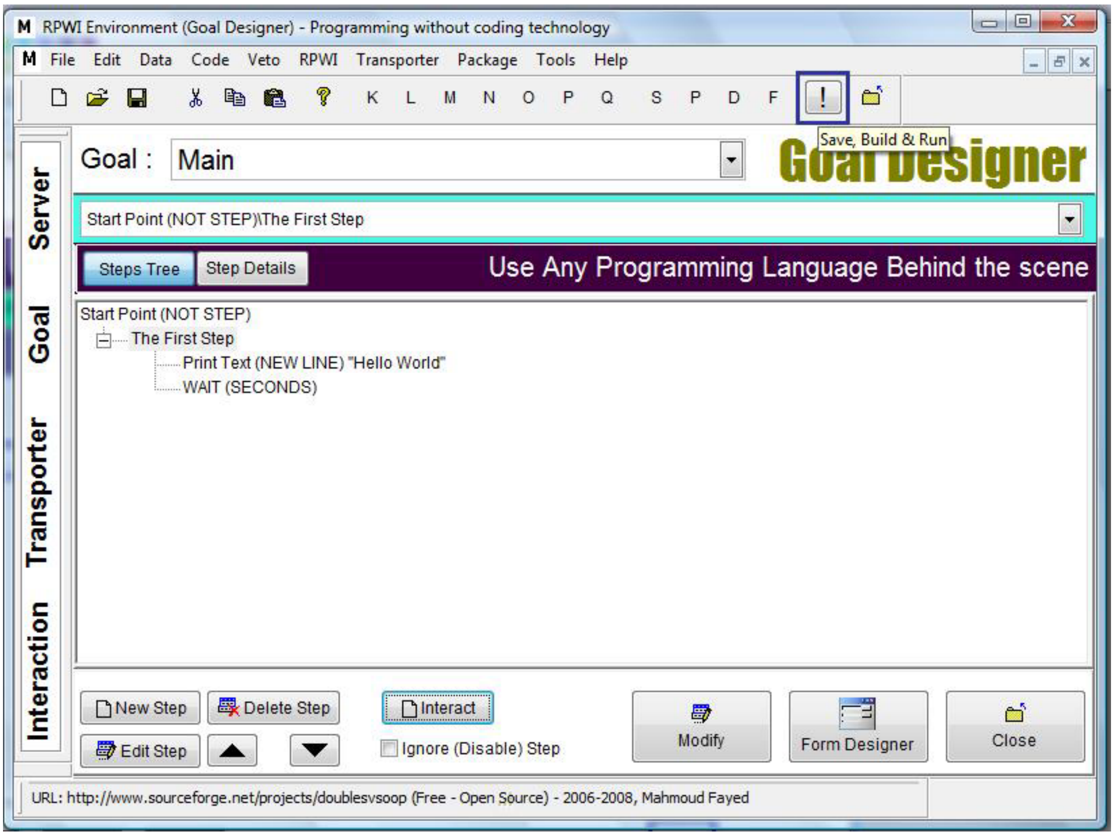
Select the first step then Click "Interact" button



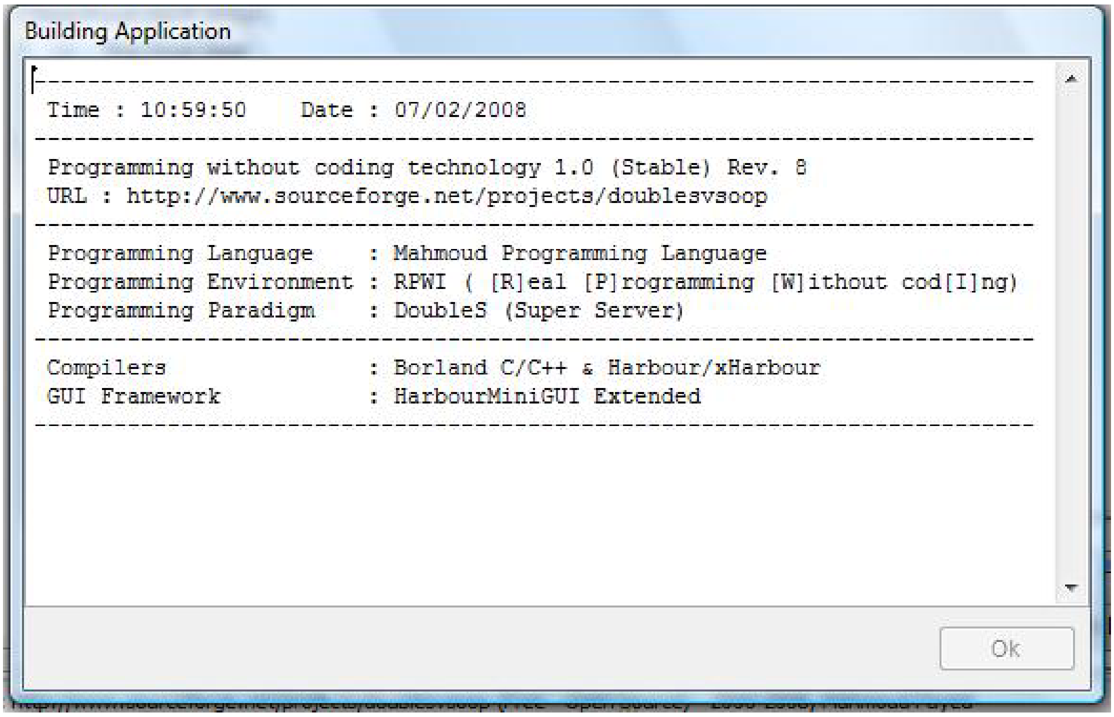
Select Domain (Console Applications) then Select Component (Wait Key/Seconds) then Click "ok" button



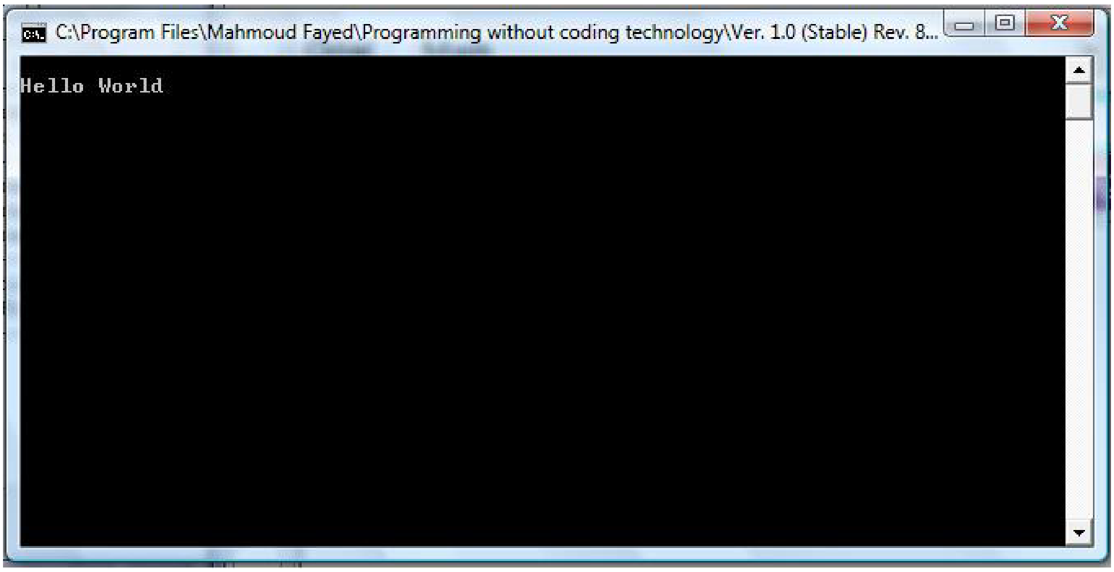
Check "wait n seconds" then determine the number of seconds to wait them click ok



Run the program



Wait for saving & building the application



Have fun seeing the application in the runtime

Setting Colors & Clearing Screen:-

Simply we can set the colors of our application

The **standard** colors used for all screen output, while the **enhanced** colors used for Getting input and highlights while the **unselected** colors used for the unselected input lines

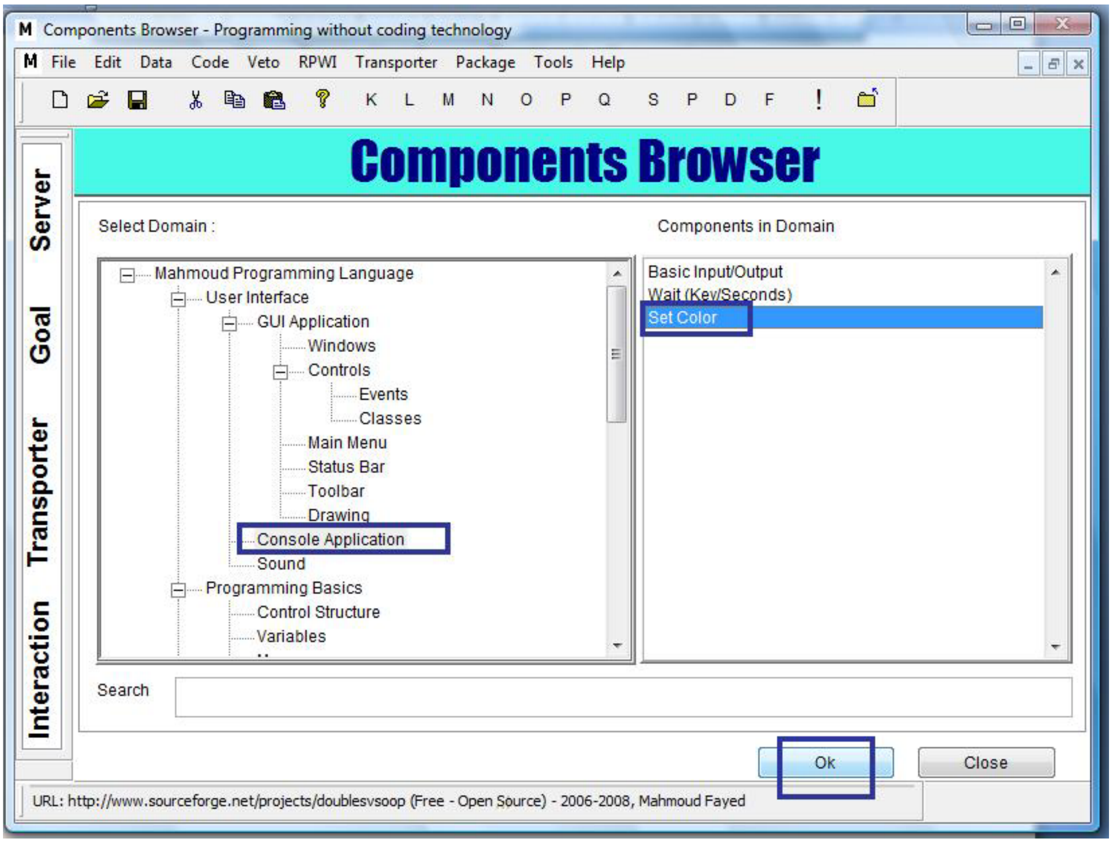
Available Colors:-

Black	Blue	Green	Cyan
Red	Magenta	Brown	White
Gray	Bright Blue	Bright Green	Bright Cyan
Bright Red	Bright Magenta	Yellow	Bright White

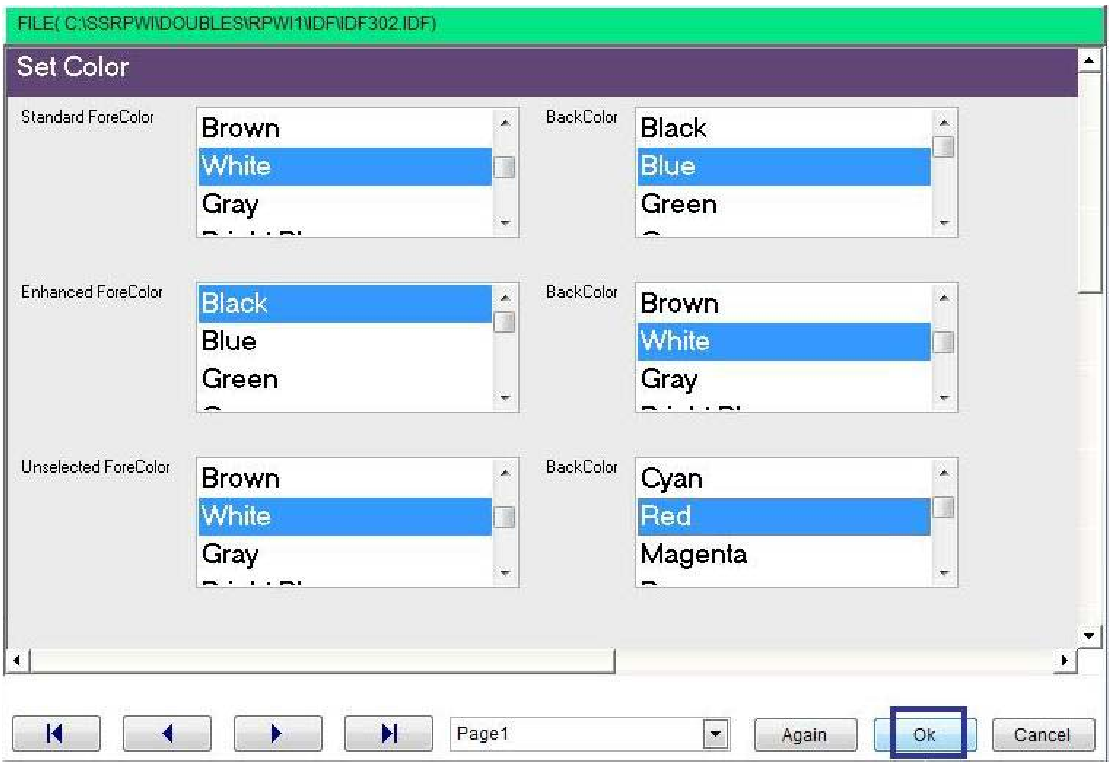
Component Address:-

- Domain (Console Application)
- Component (Set Color)

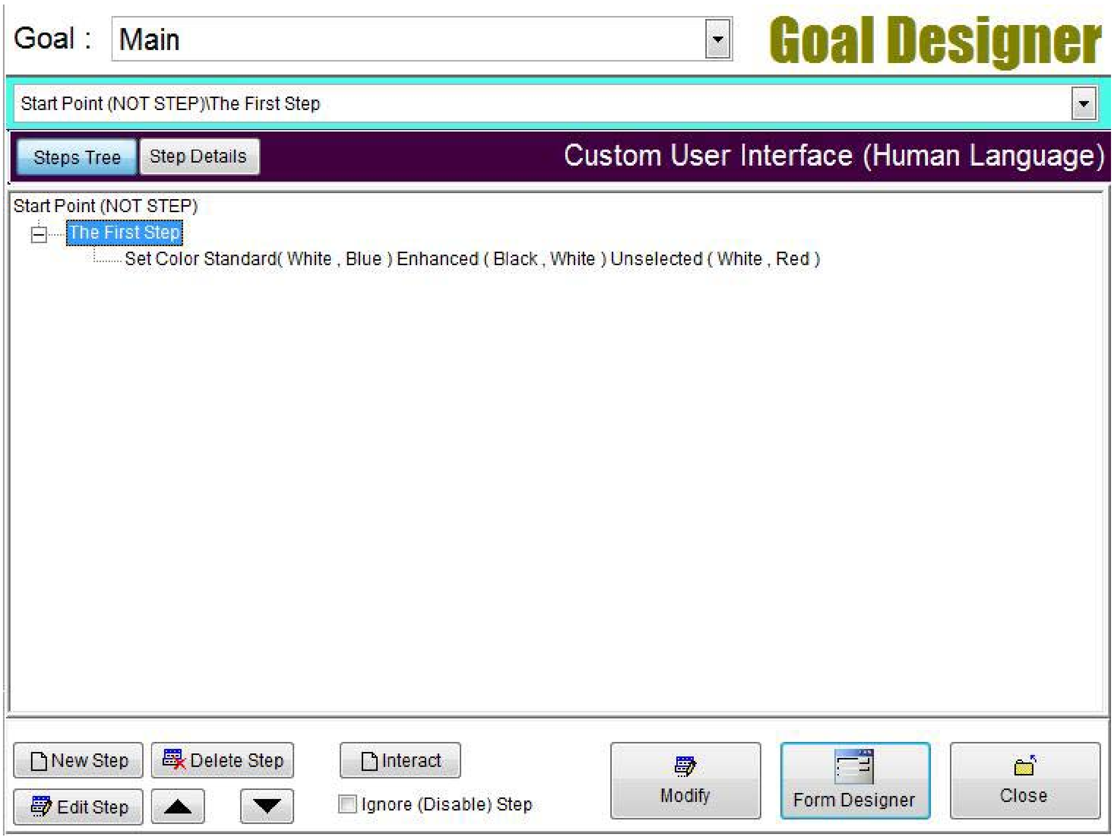
Screen Shots:-



Set color – Component address

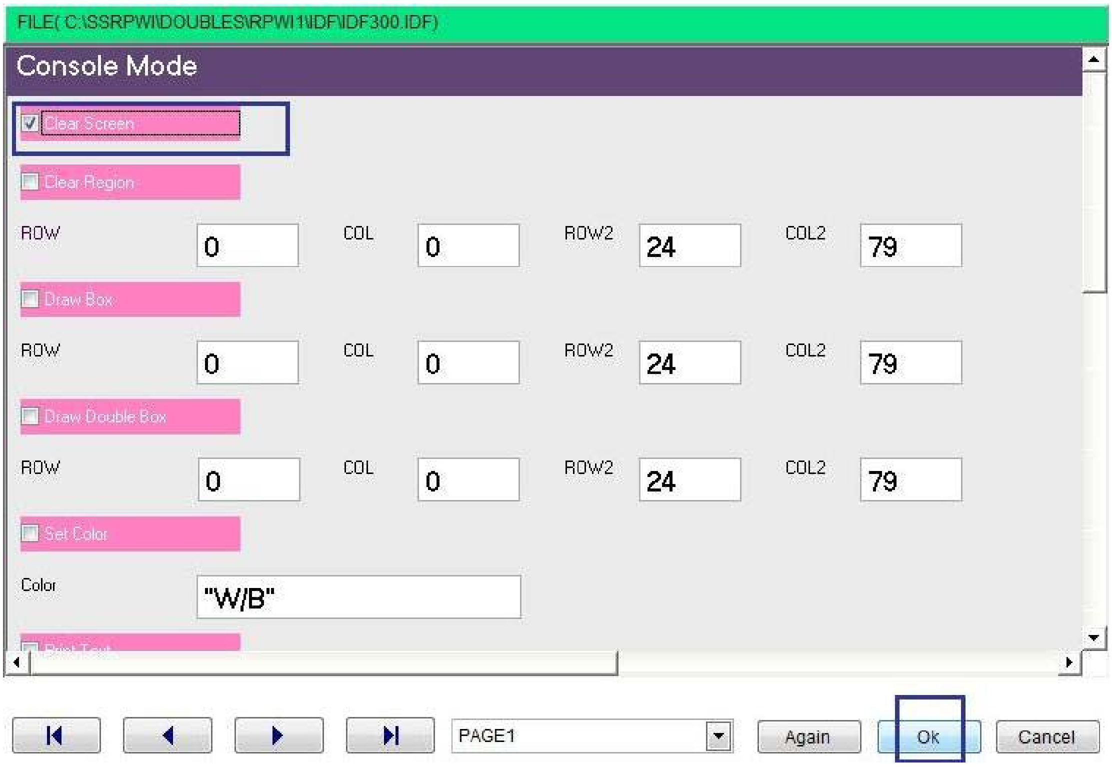


Set color – interaction page



Set color – Interaction Result

Now we are going to clear the screen using the selected color
Component Address: Domain (Console Application) Component (Basic Input/output)

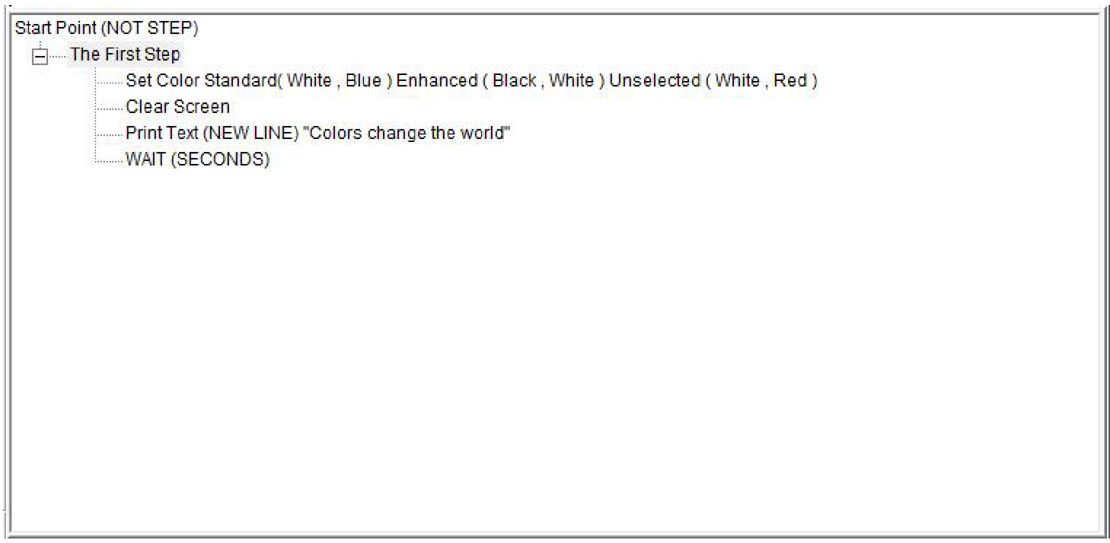


Clearing screen

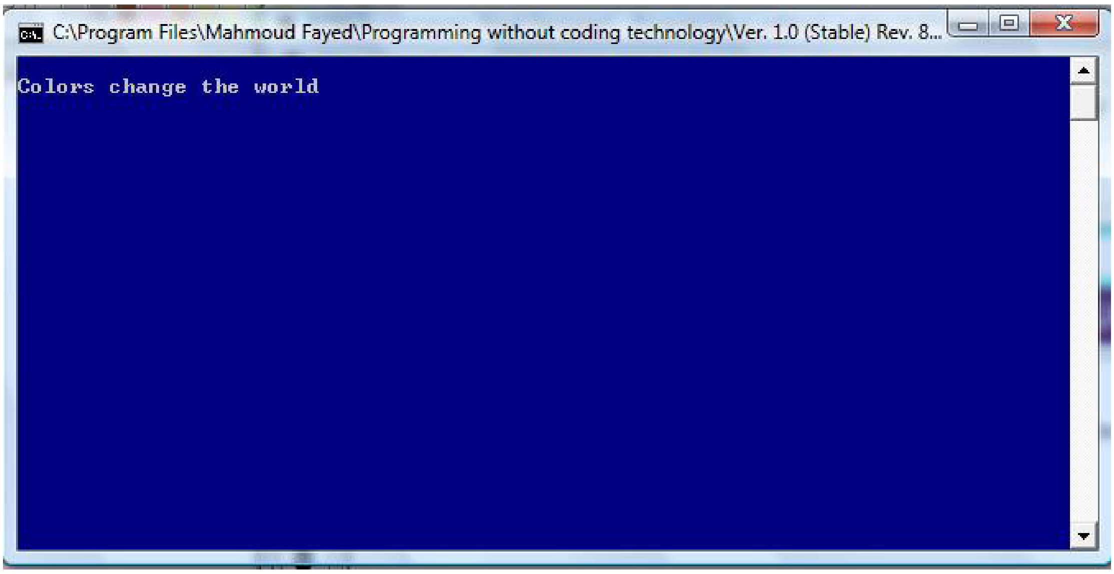


The result of interaction process

Now after selecting colors and clearing screen we can do as in the first example, printing text then waiting for number of seconds.



Our steps tree include setting colors, clearing screen, printing text & waiting



The final application

Clearing a rectangle area, drawing a box

We can clear a rectangle area on the screen by filling it with space characters and using standard color setting

Also we can draw a box around a rectangle area.

Component Address:

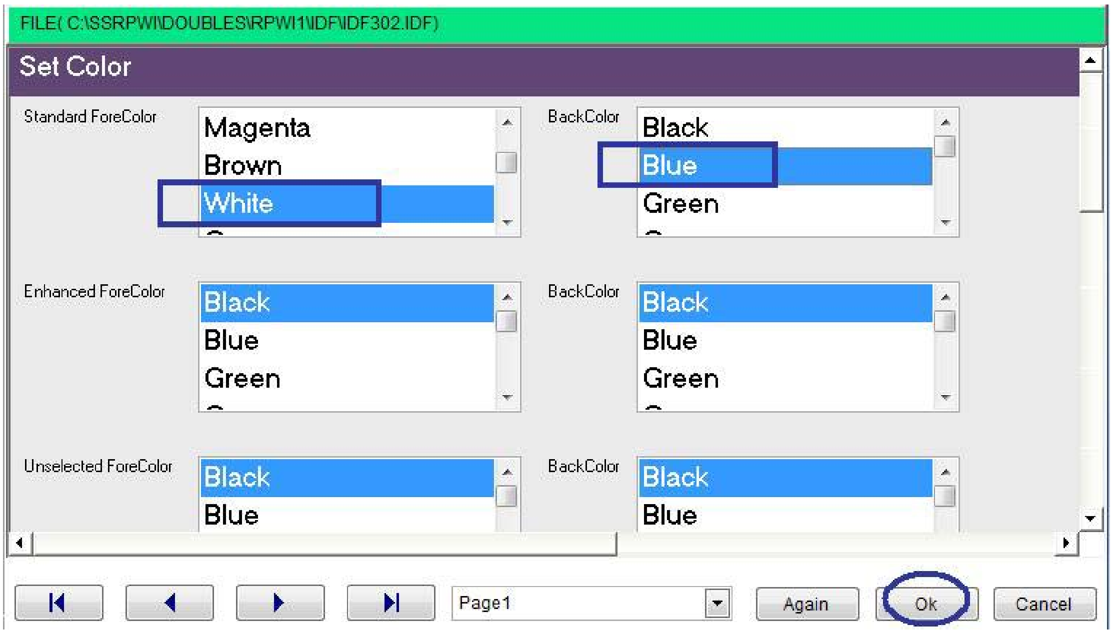
- Domain Name (Console Application)
- Component Name (Basic Input/output)

Example:

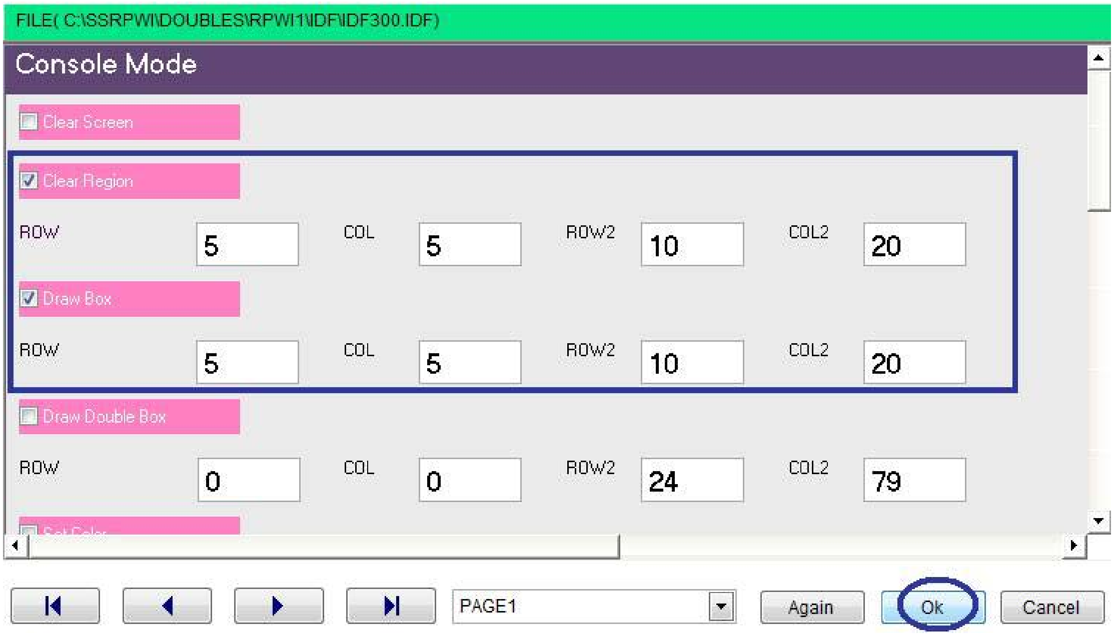
We are going to clear a rectangle area using the blue color
Then drawing a box around this area using the white color

Row/Top	5
Column/Left	5
Bottom	10
Right	20

Screen Shots:-

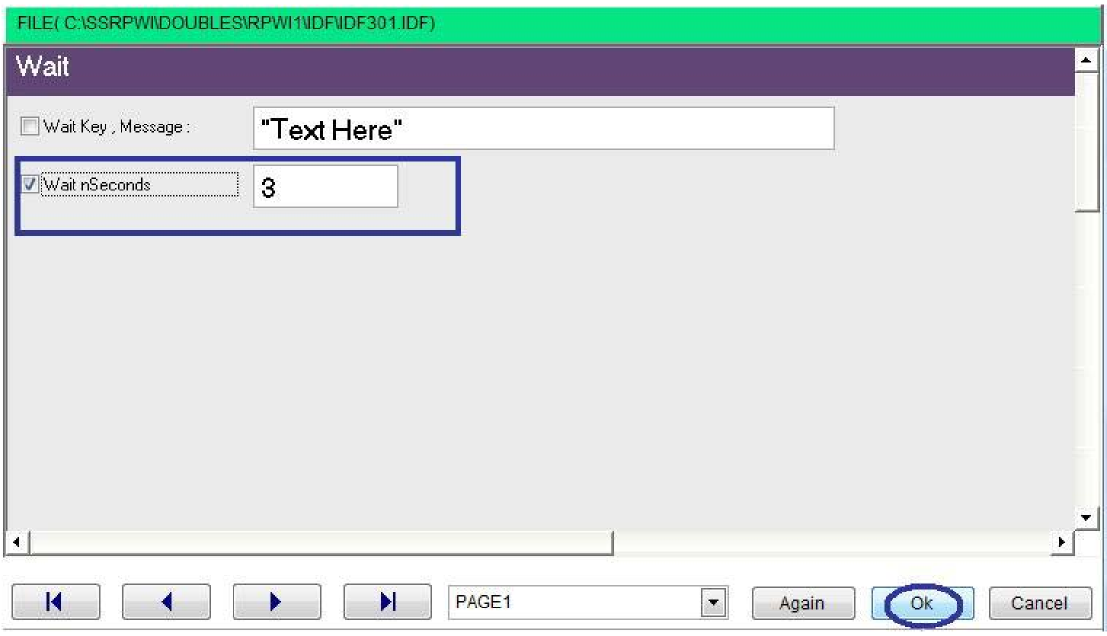


Setting colors

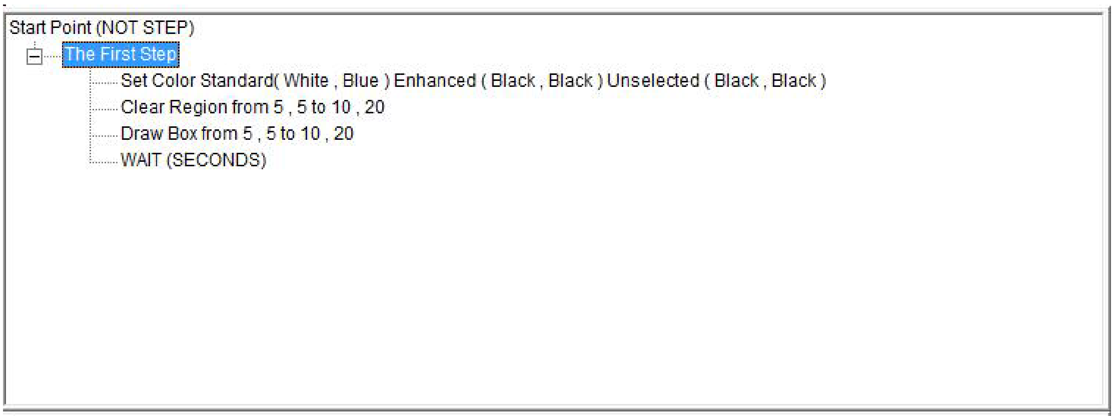


Clearing the rectangle region then drawing a box

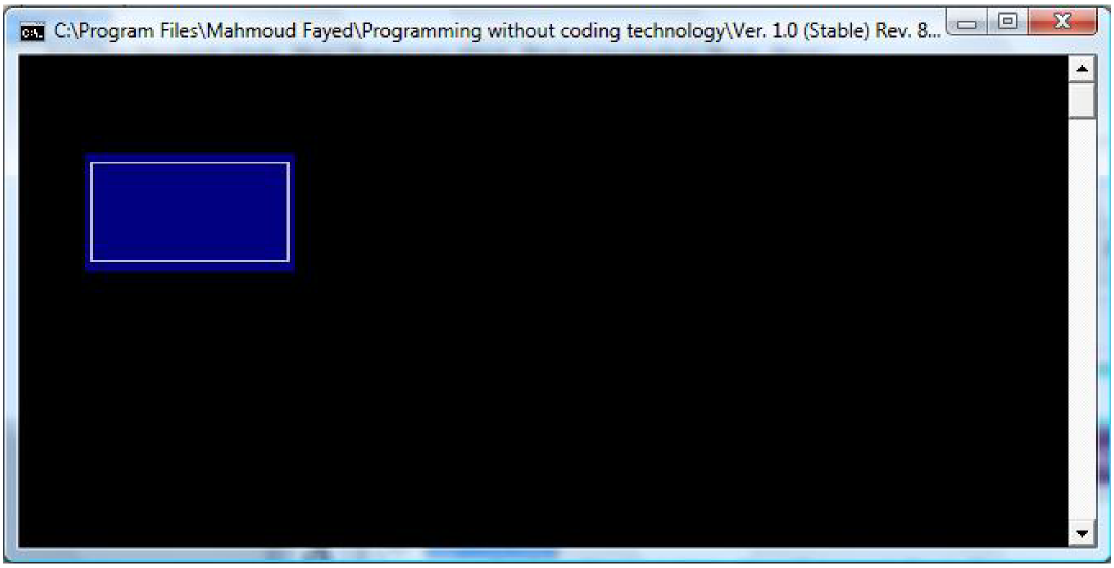
`



Waiting for number of seconds



The final steps tree



The final application

Variables Assignment:-

All types can be assigned to named variables Named variables can be 1 to 63 characters long identifiers, starting with [A-Z|_] and further consisting from additional [A-Z|0- 9|_] characters, up to 63 characters, which are not case sensitive.

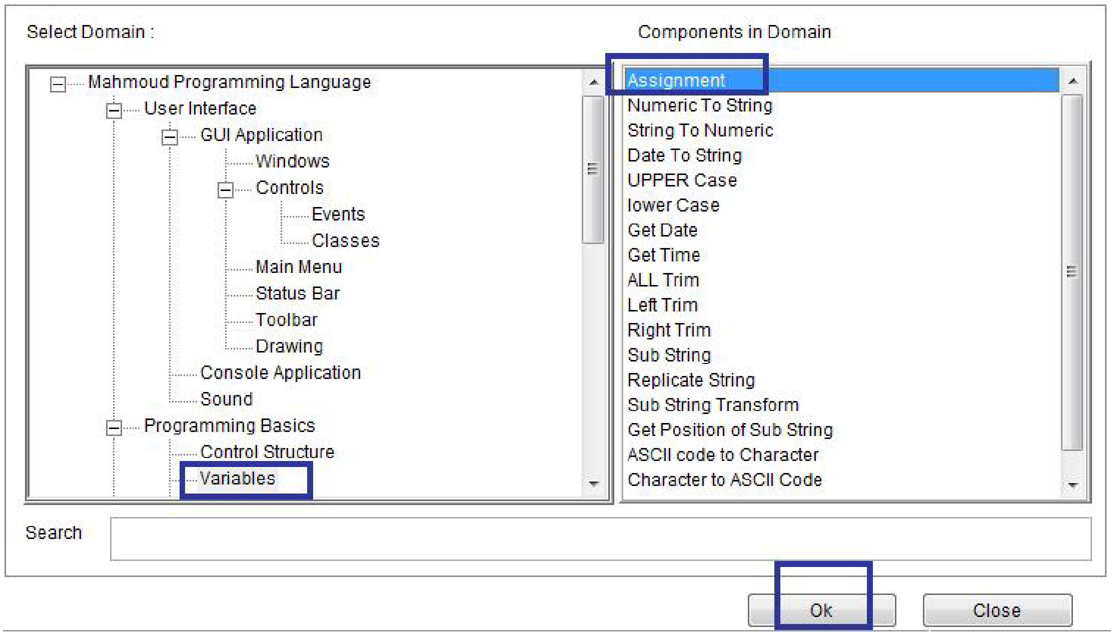
Assignment is not more than
Left Side (Variable Name) = Right Side (Value)
Like in Math

Component Address:-
Domain (Variables)
Component (Assignment)

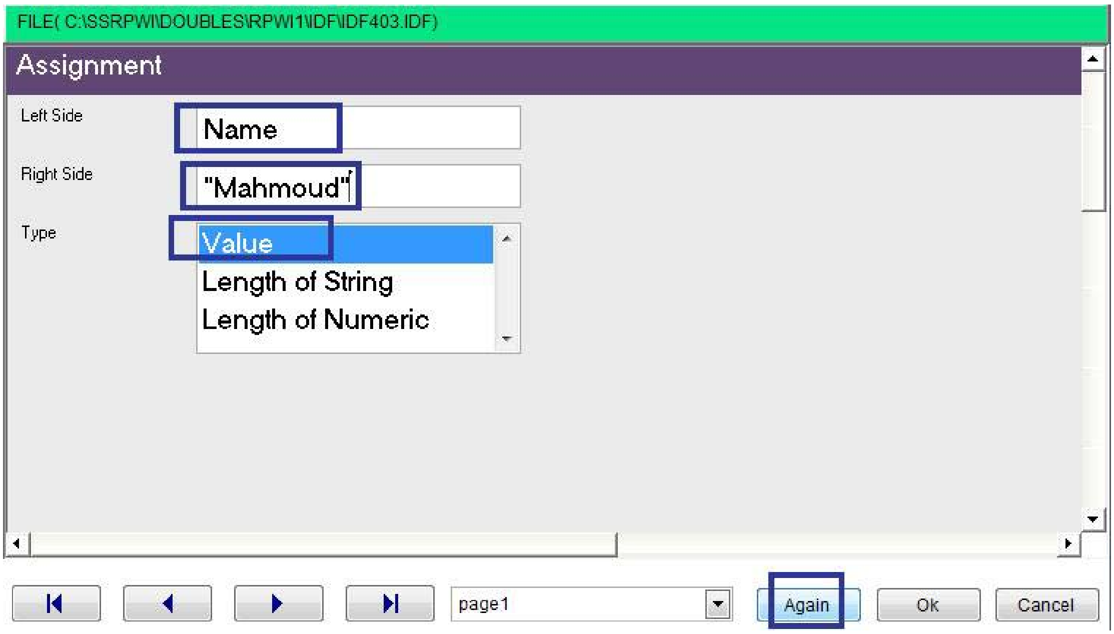
Example:-

We declare two variables the first is (Name) and its value is (Mahmoud)
While the second are (Age) and its value is (21).

Then we display the values of the variables
As we do in displaying a string on the screen
But variables names written directly without double coition

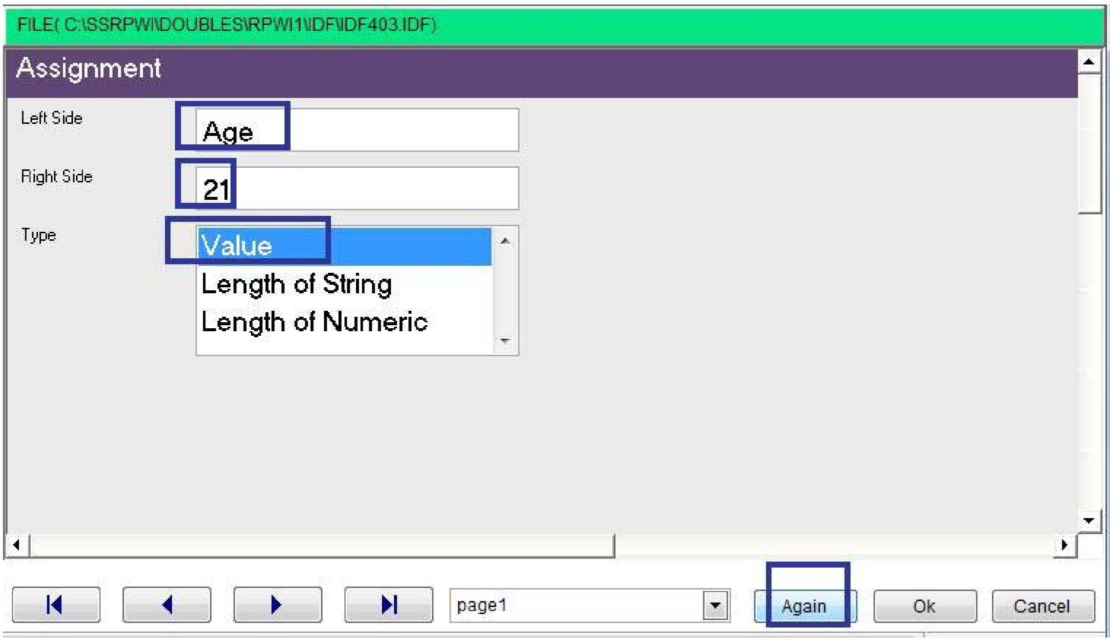


Domain (Variables) – Component (Assignment)



Interaction Page of the assignment component

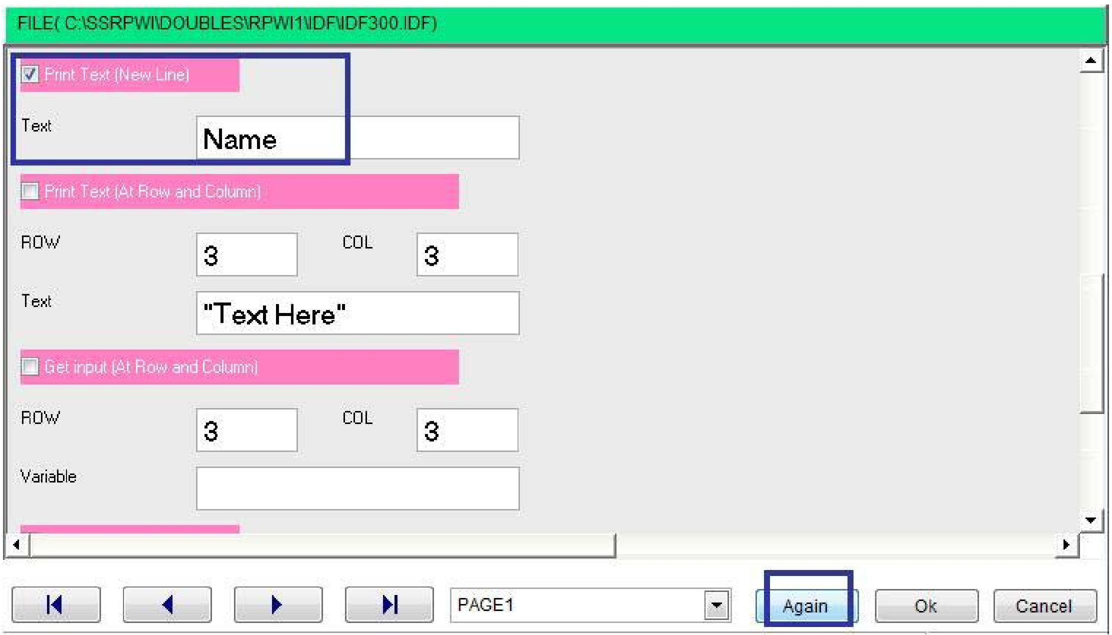
Since we need to use the same component more than one time
Then we click "Again" button instead of clicking "Ok" button



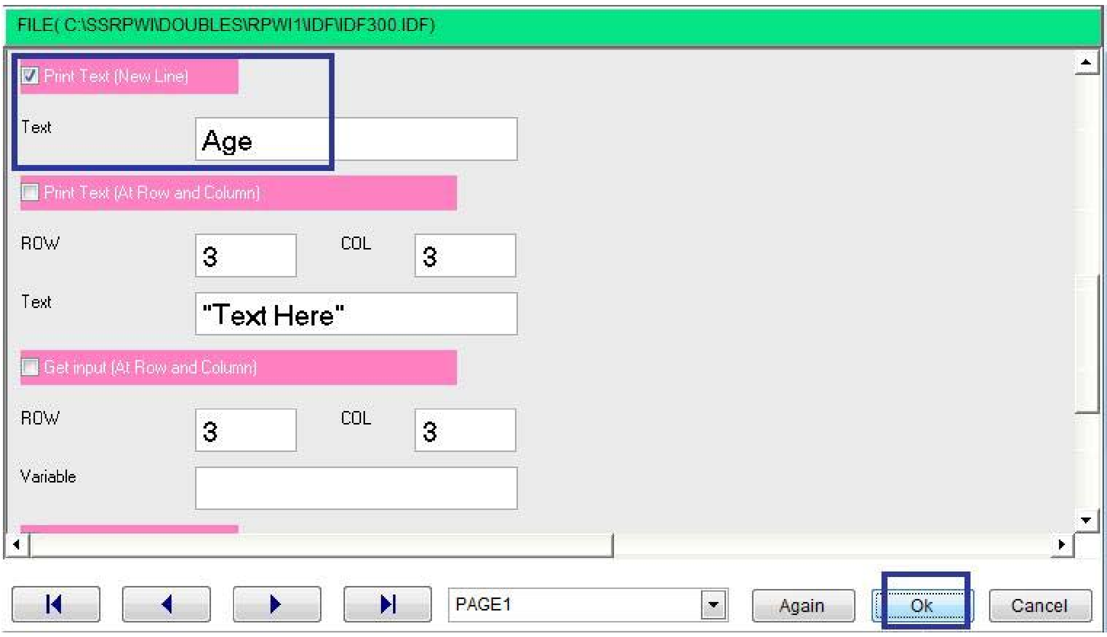
Interaction Page of the assignment component



The result of the interaction



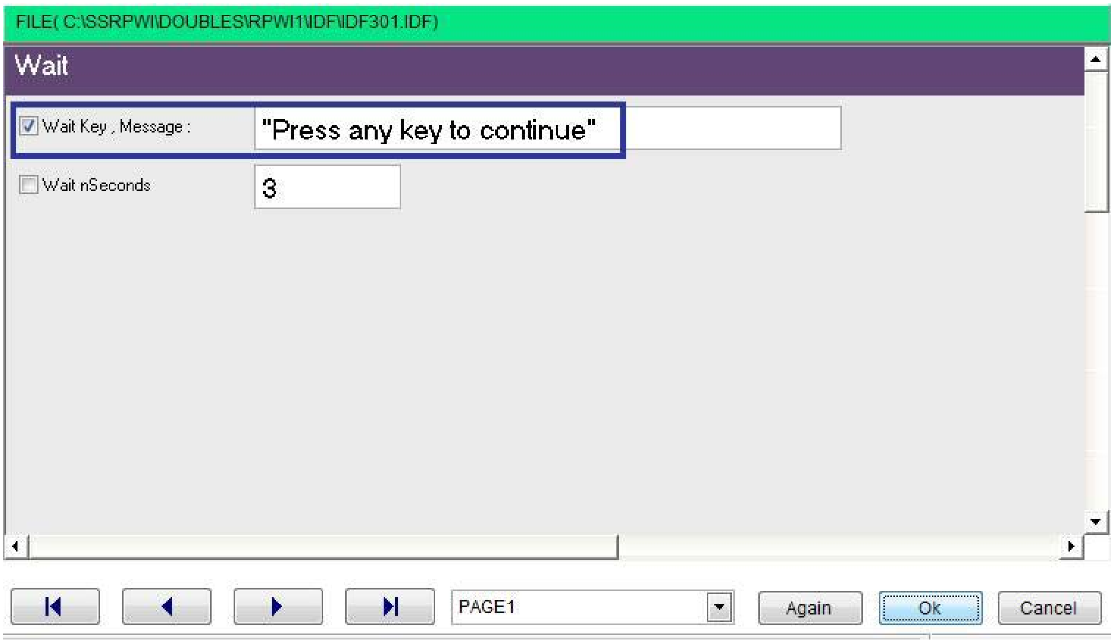
Interaction Page



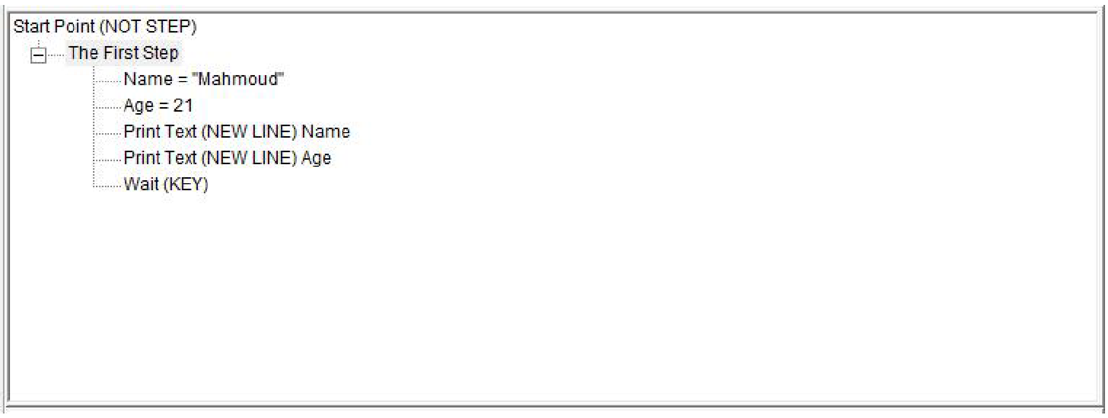
Interaction Page



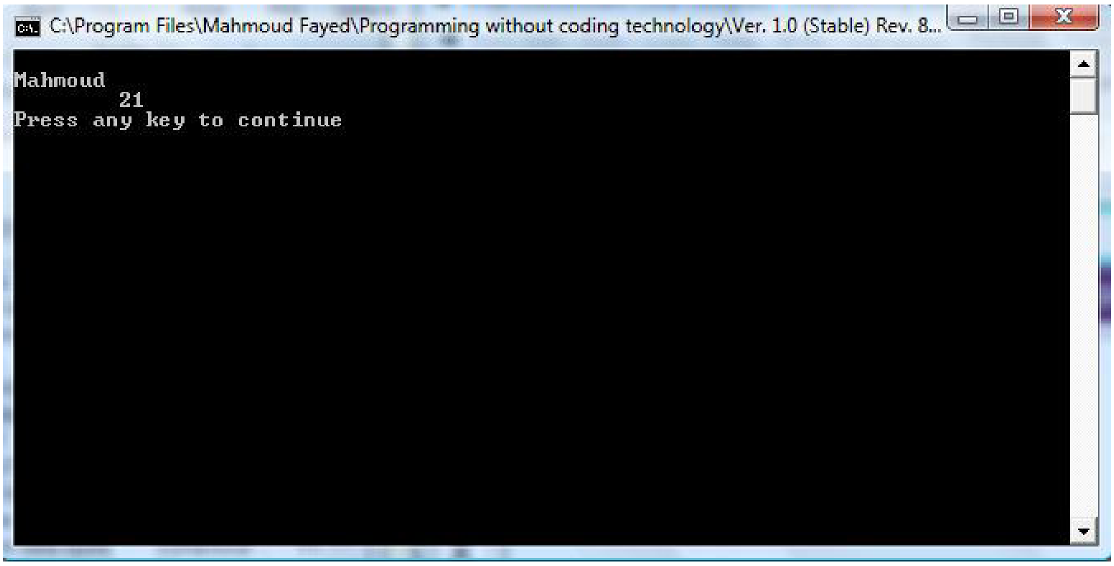
Interaction Page



Interaction Page



The final – Steps Tree



The final application

Strings:-

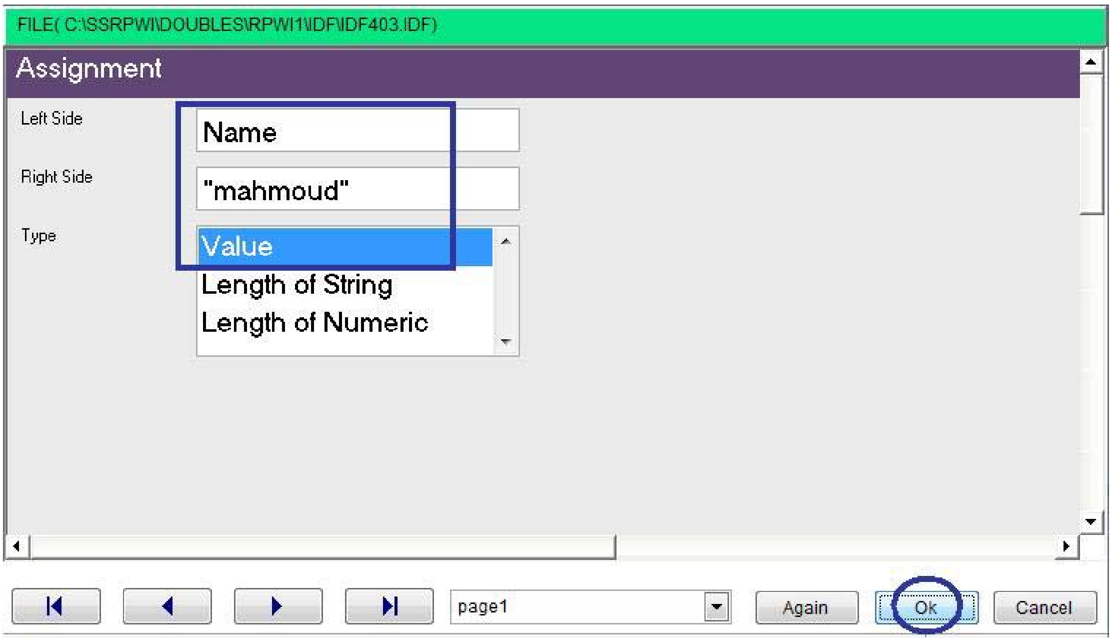
Component Address:

- Domain (Variables)
- Components
  - UPPER case
  - lower case
  - All trim
  - Left trim
  - Right trim
  - Sub String
  - Replicate String
  - Sub String transform
  - Get position of substring
  - Get String/Variable/Array Length

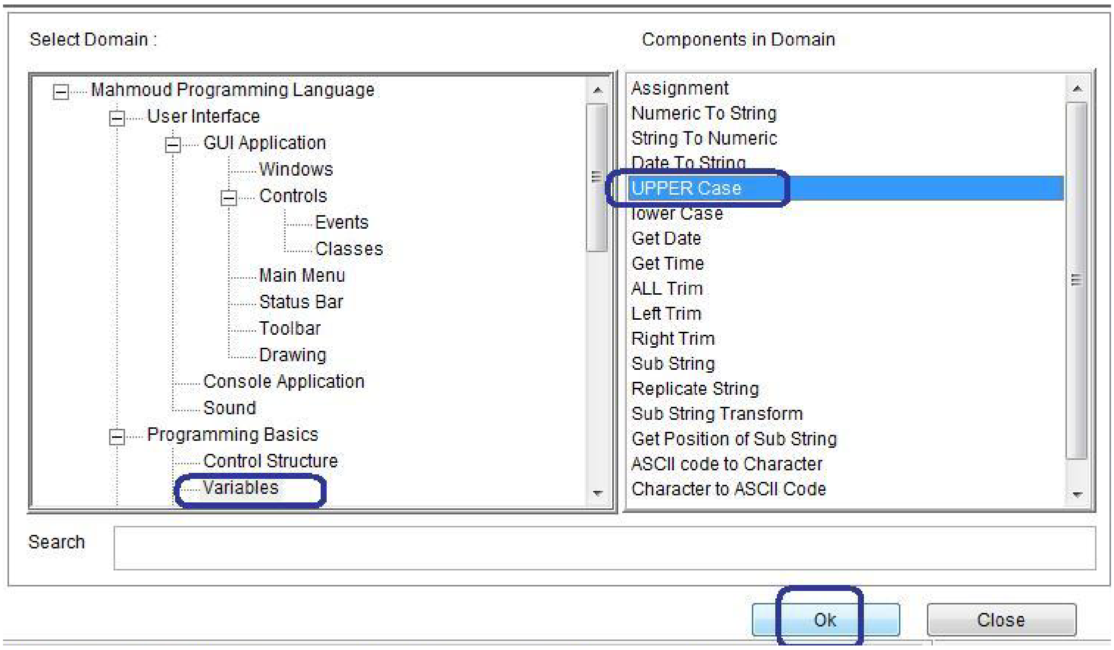
Upper Case:

Returns a copy of the string with all alphabetical characters converted to uppercase
All other characters remain the same as in the original string

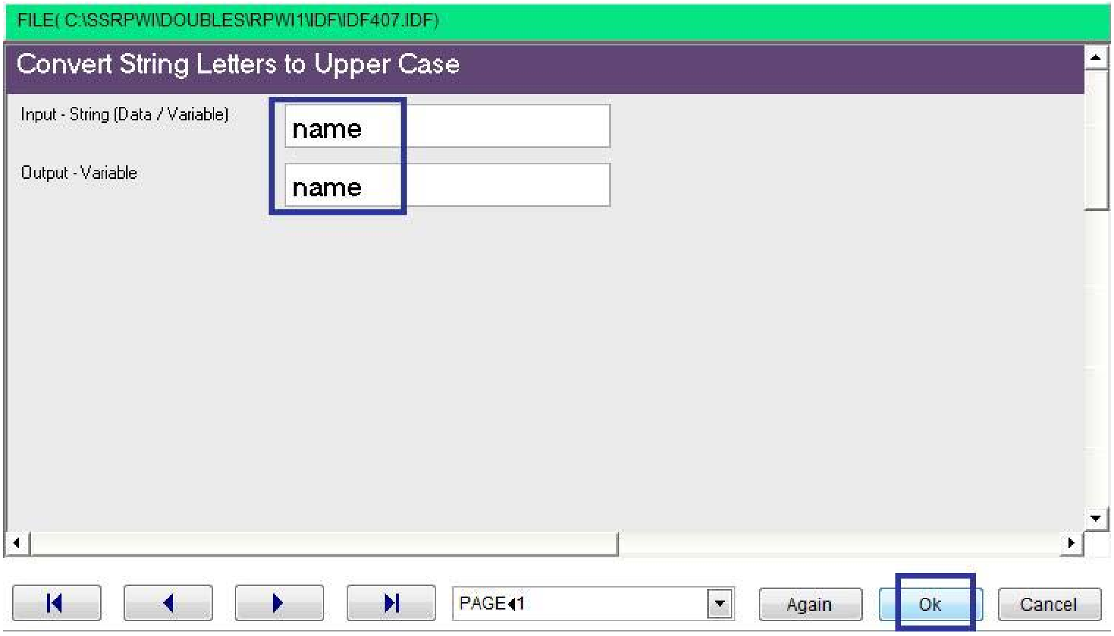
Example – Screen Shots:-



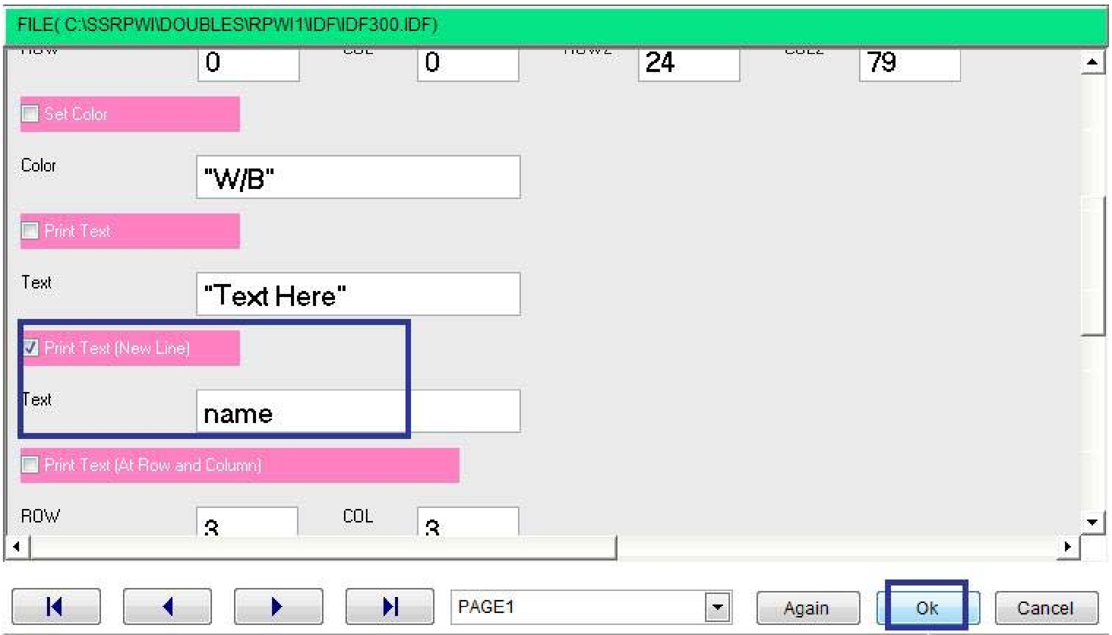
Setting variable Name = "mahmoud"



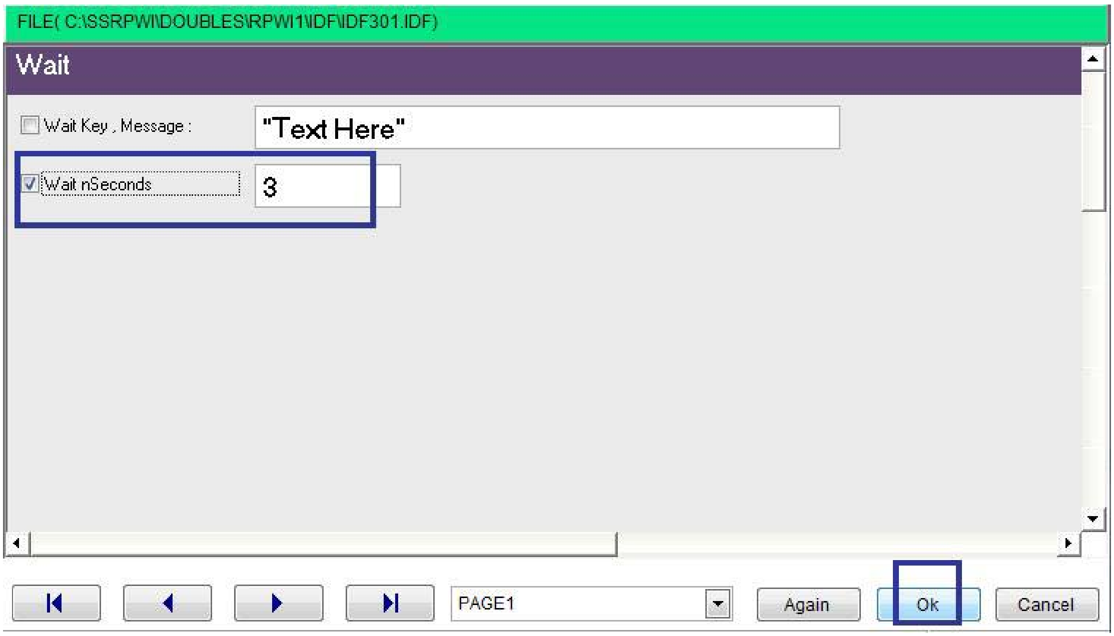
Domain (Variables) – component (UPPER Case)



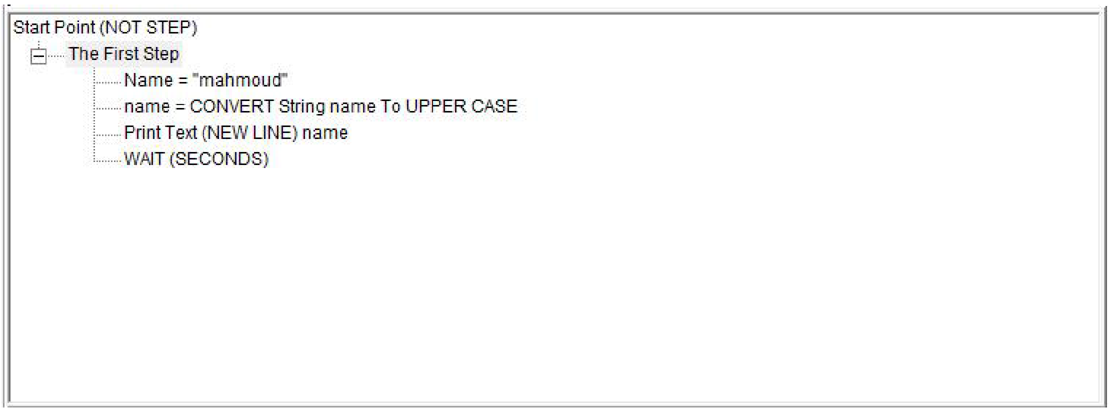
input variable name = output variable name = name



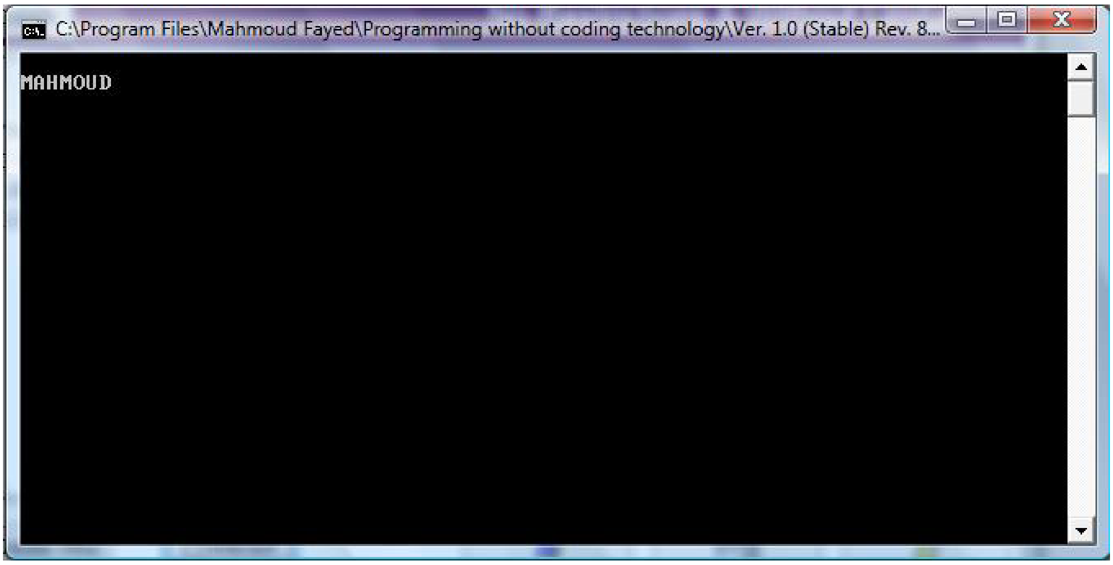
Printing the value of the variable name on the screen



Waiting for 3 seconds



The final steps tree

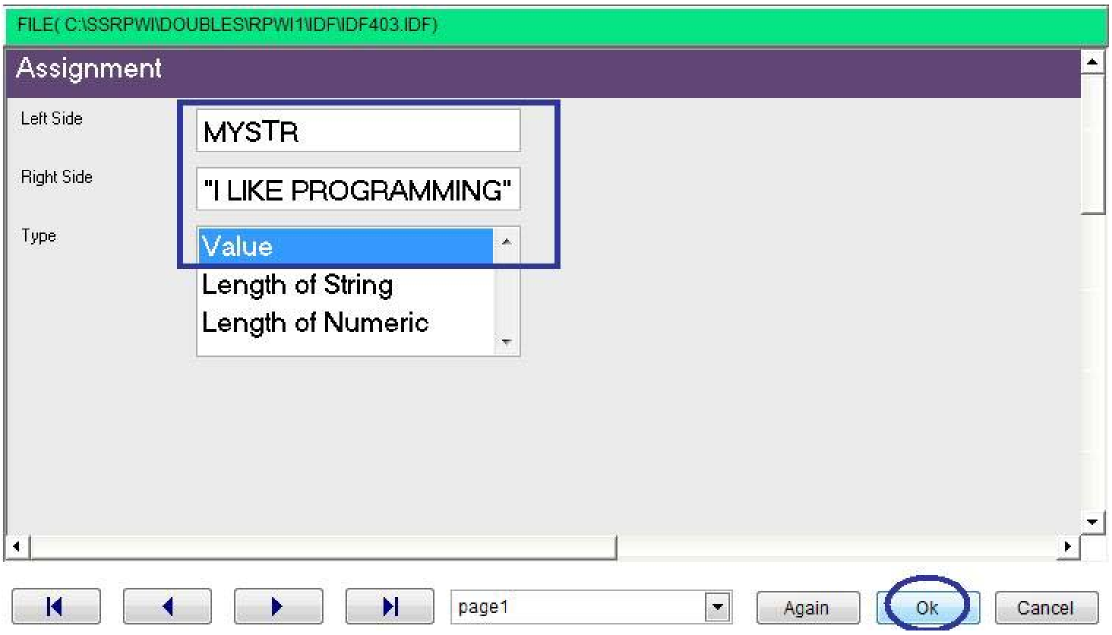


The final application

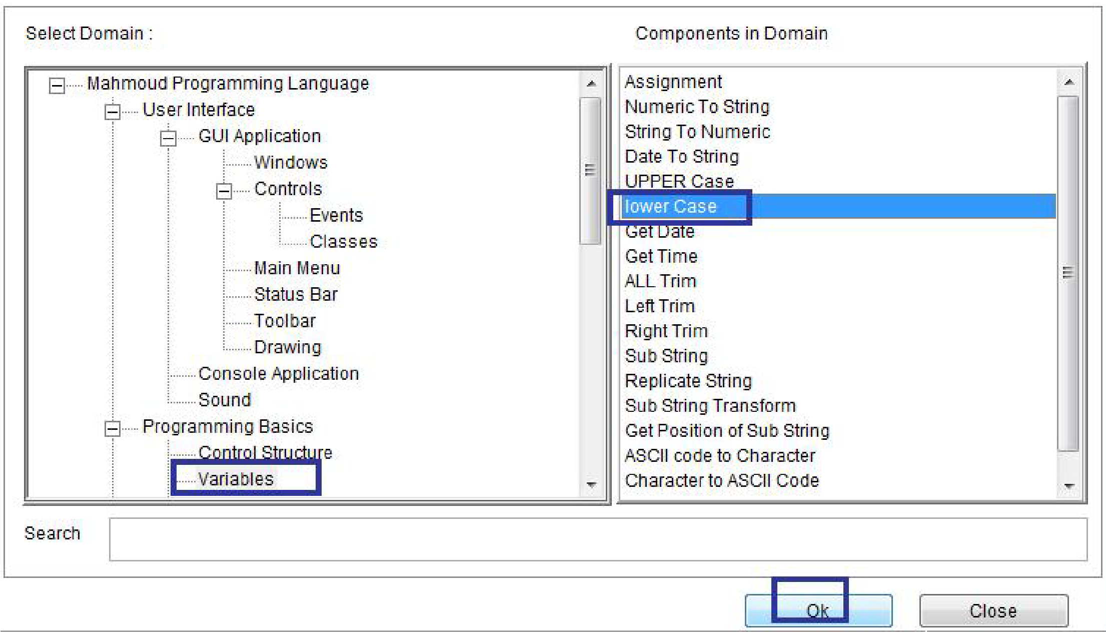
Lower case:

Returns a copy of the string with all alphabetical characters converted to lowercase
All other characters remain the same as in the original string

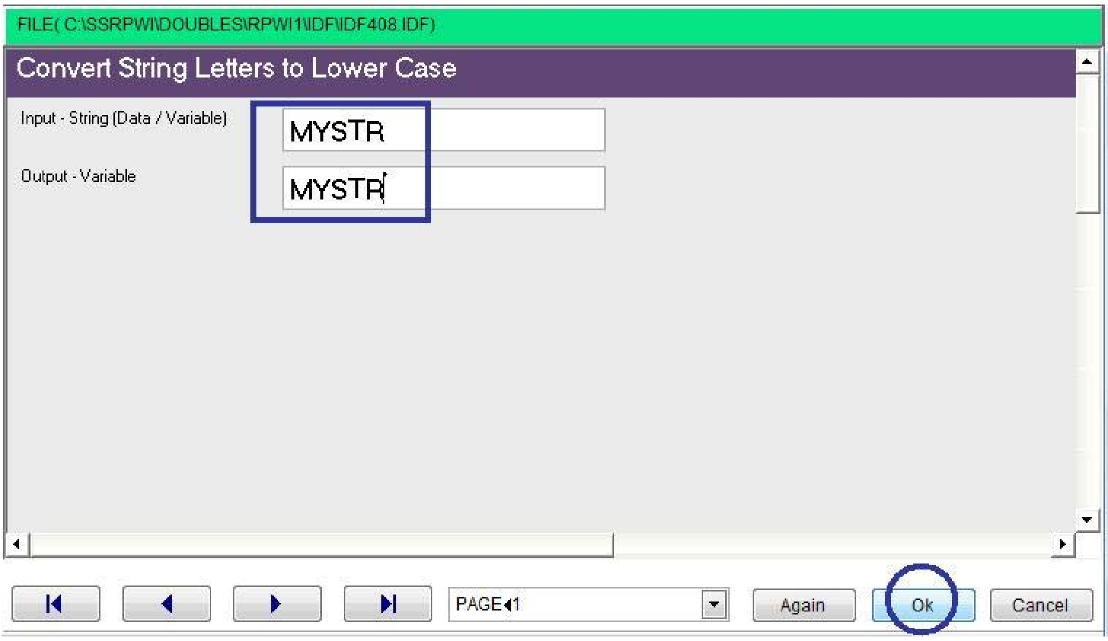
Example – Screen Shots:-



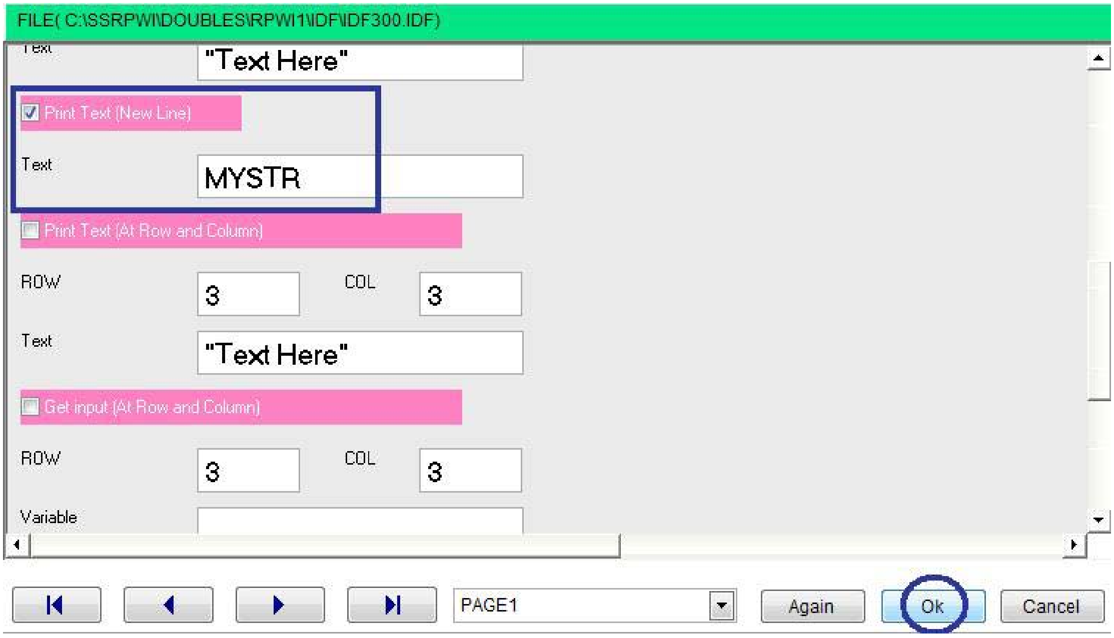
declare variable mystr = "I LIKE PROGRAMMING"



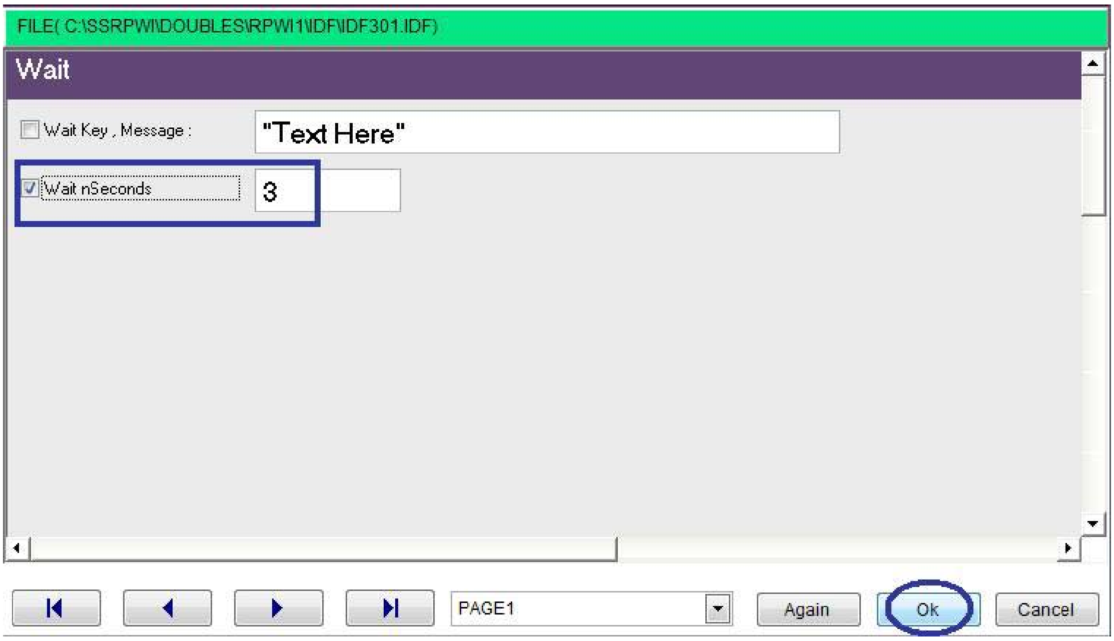
Domain (Variables) Component (lower case)



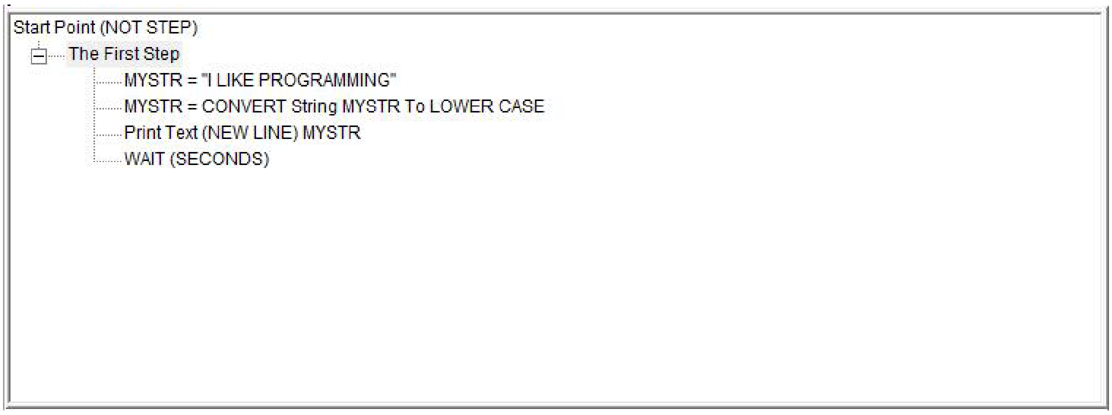
Interaction page – input variable name = output variable name = MYSTR



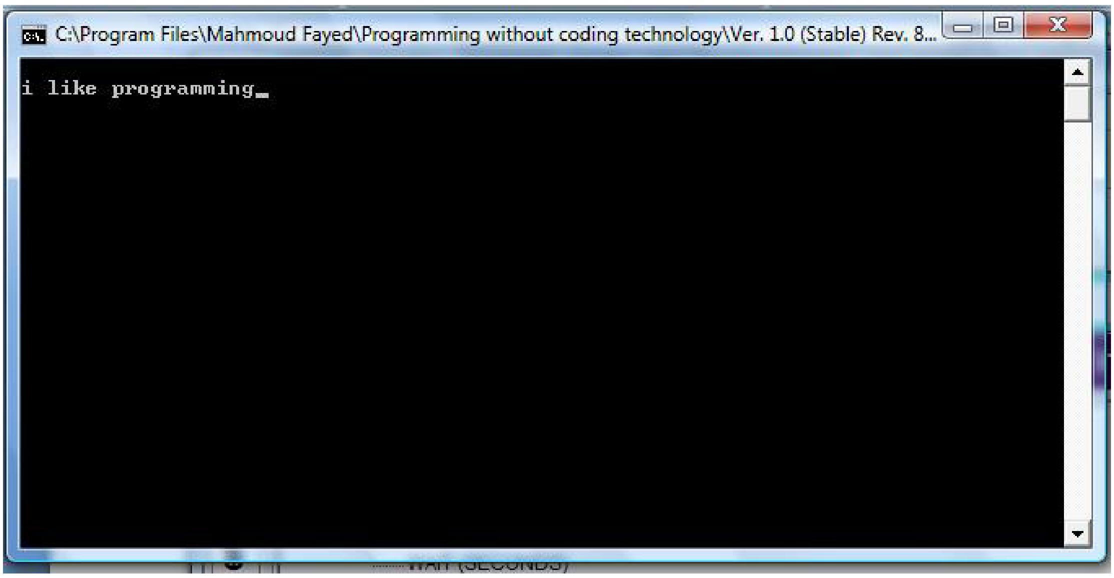
Display the value of the variable mystr on screen



Wait for 3 seconds



The final steps tree

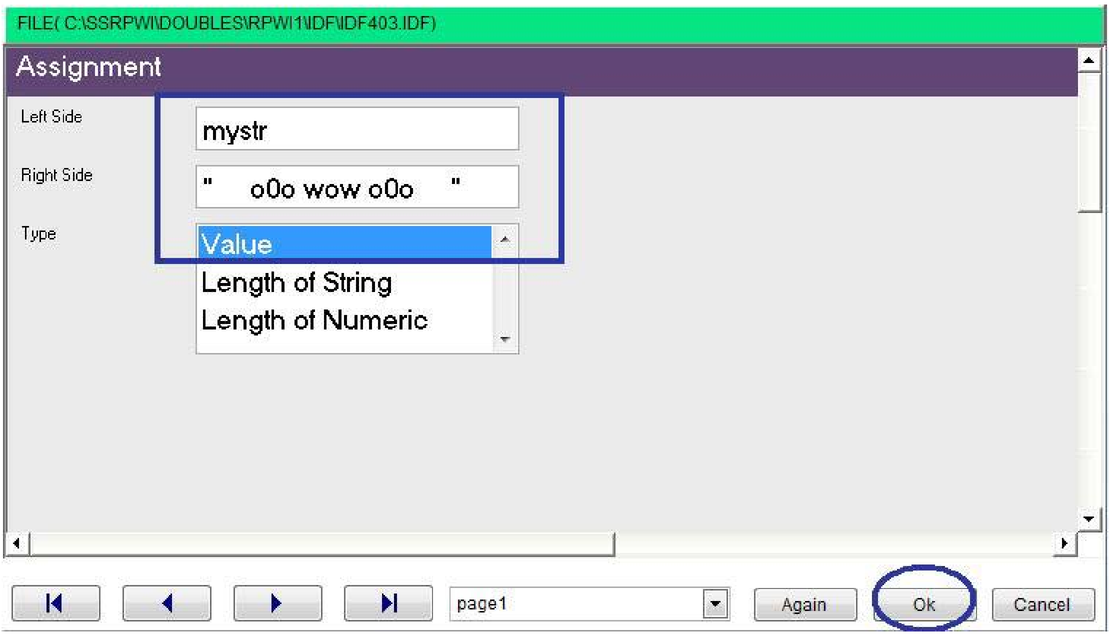


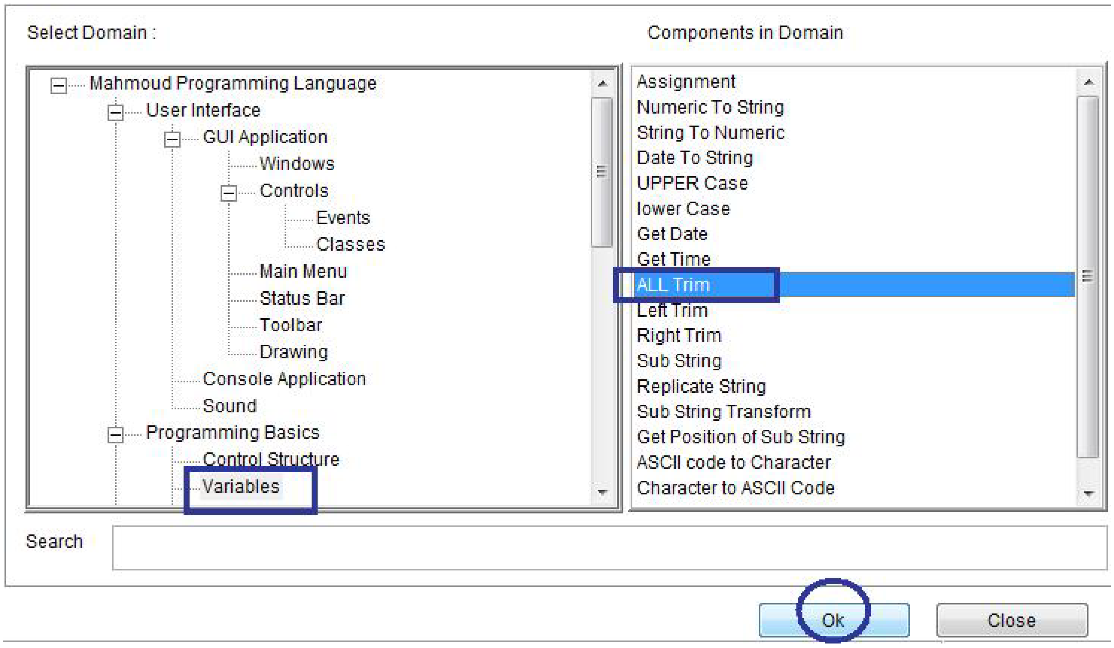
The final application

All trim:-

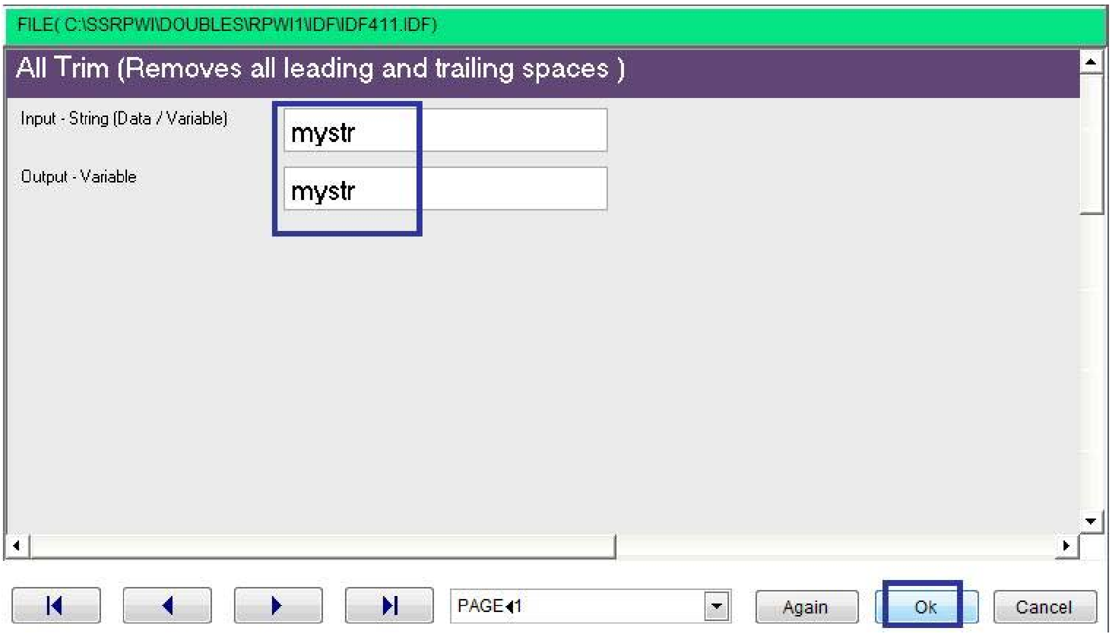
Returns the input string with all leading and trailing blank spaces removed

Example - Screen Shots:-

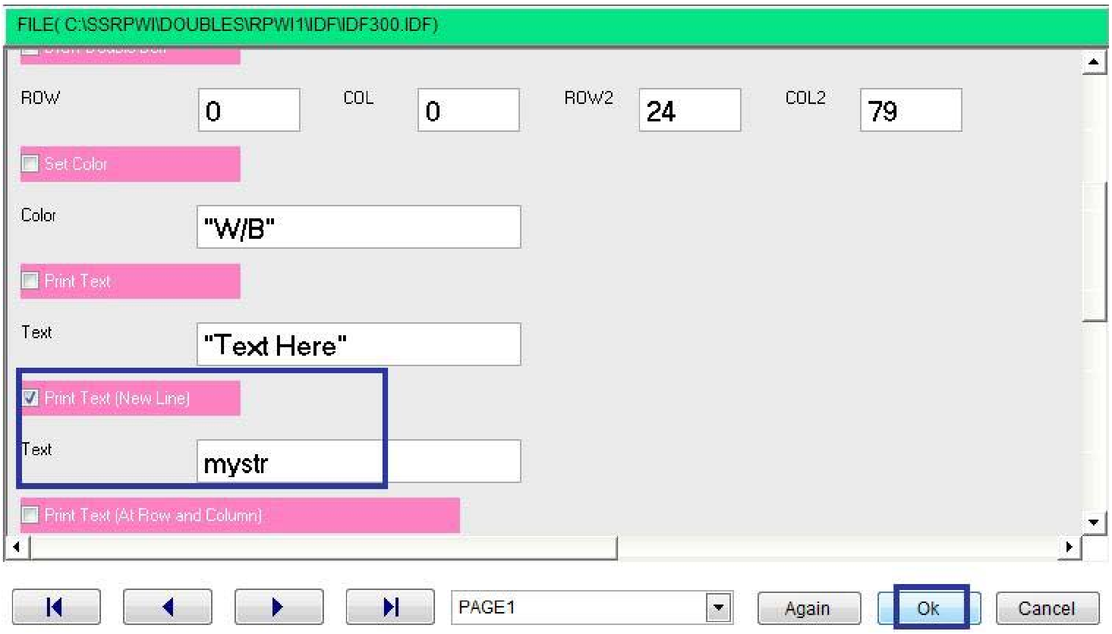




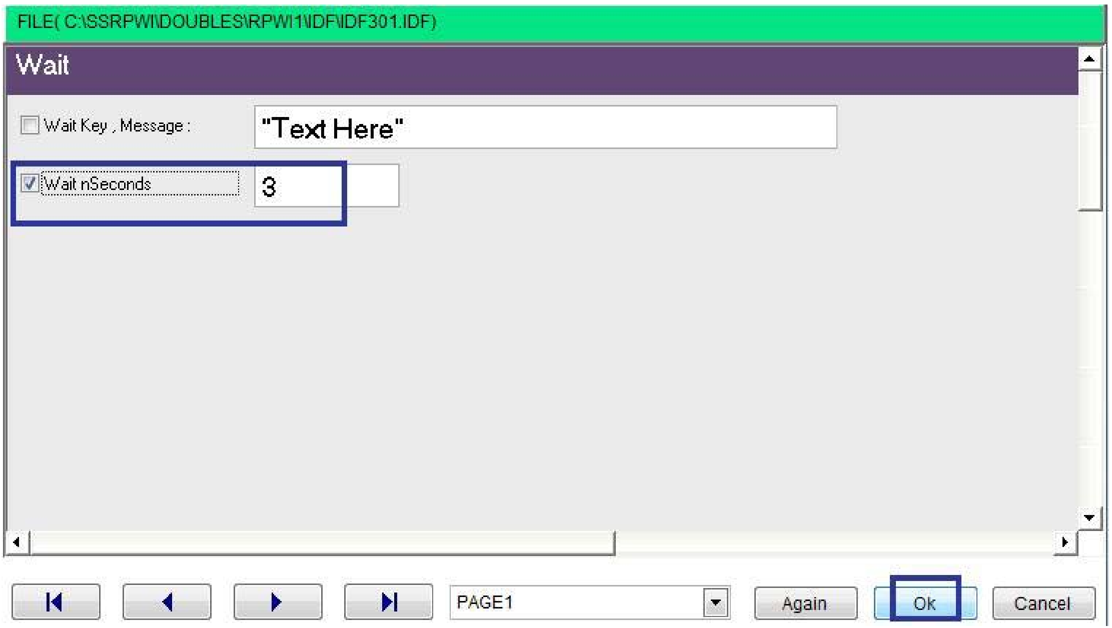
Domain (Variables) – Component (All trim)



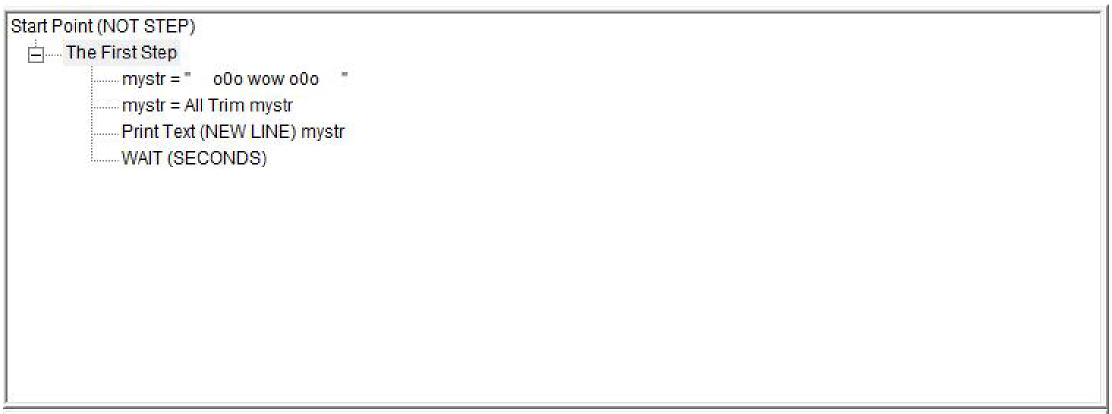
Input variable name = output variable name = mystr



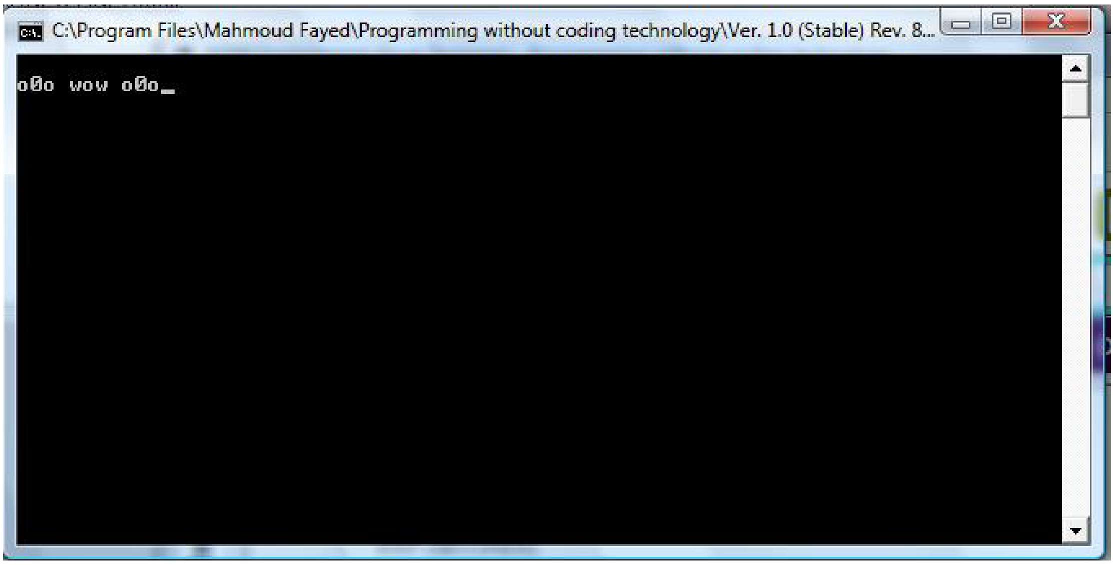
Display the value of the variable mystr on the screen



Wait for 3 seconds



The final steps tree



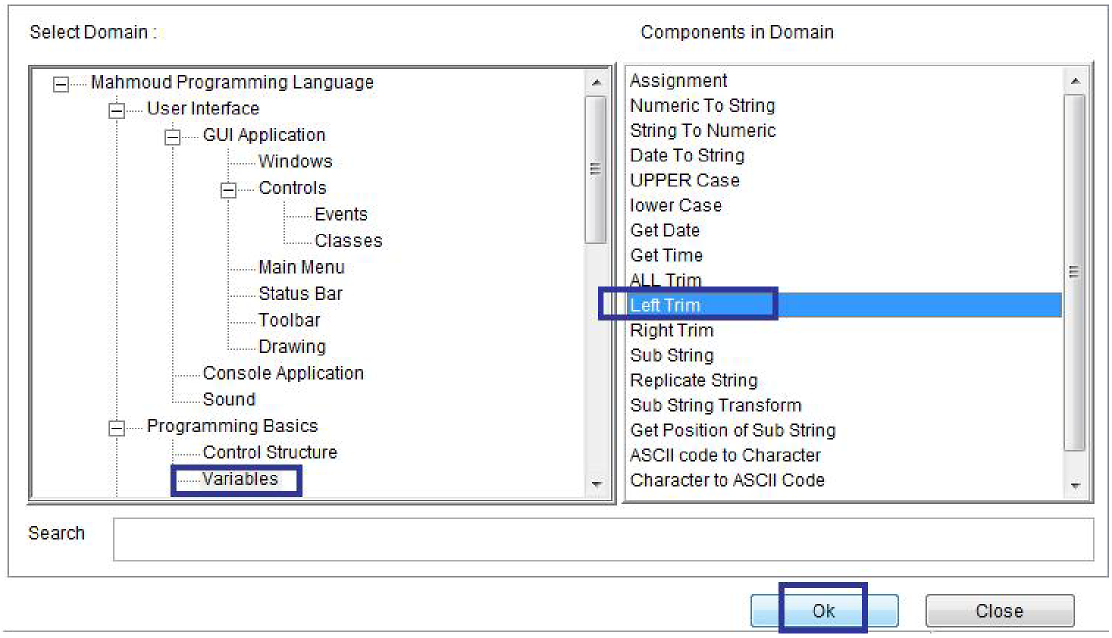
The final application

`

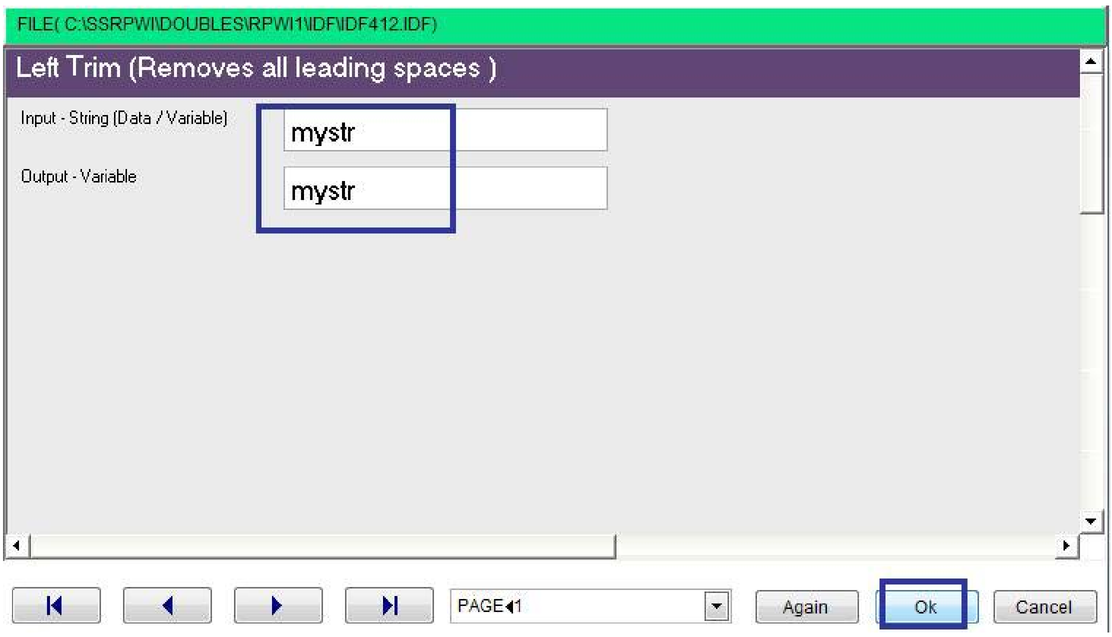
Left trim:-

Returns the input string with all leading blank spaces removed

Example – Screen shots:-



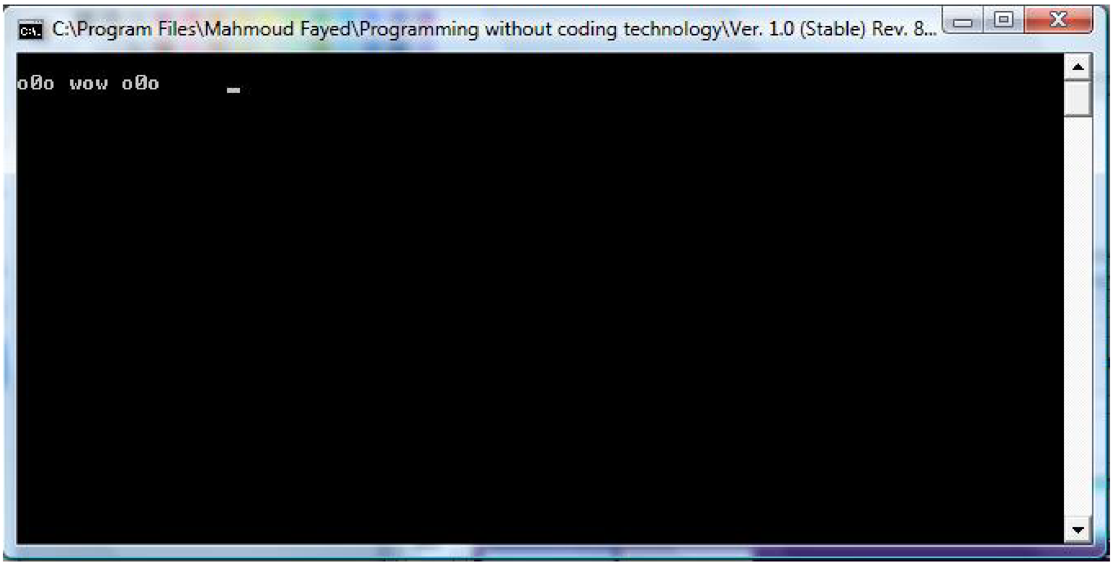
Domain (Variable) Component (Left Trim)



Input variable name = output variable name = mystr



The final steps tree

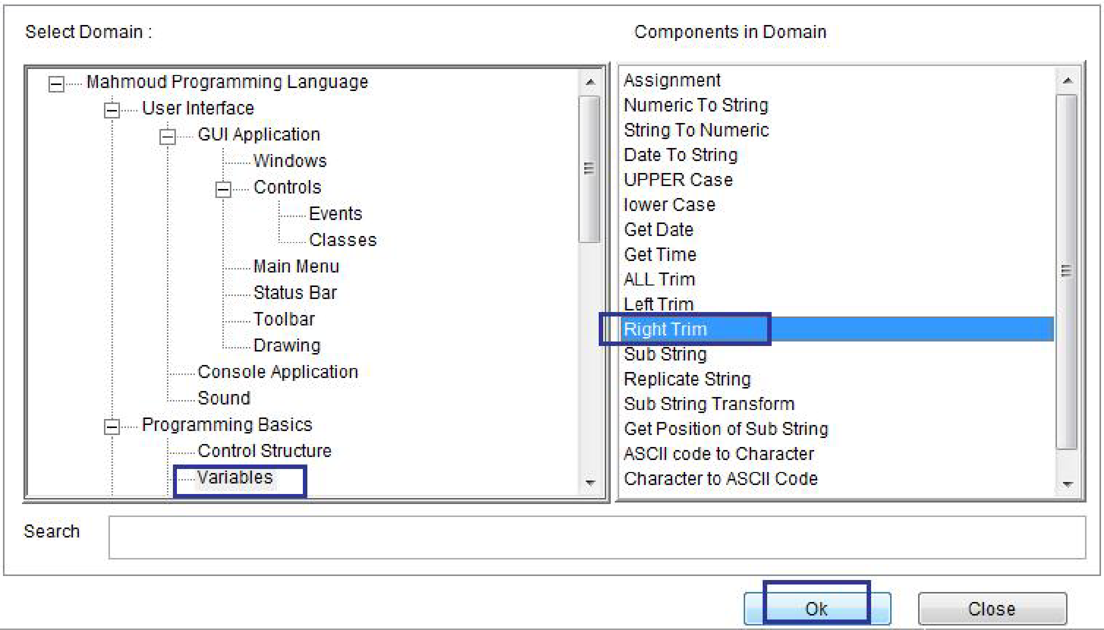


The final application

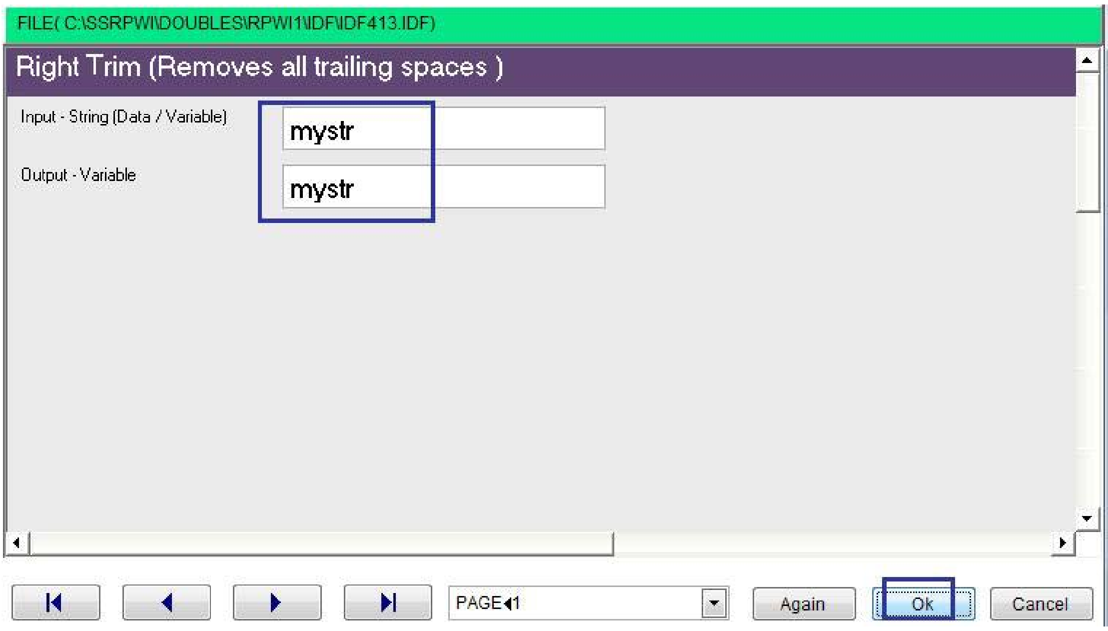
Right trim:-

Returns the input string with all trailing blank spaces removed

Example - Screen Shots:-



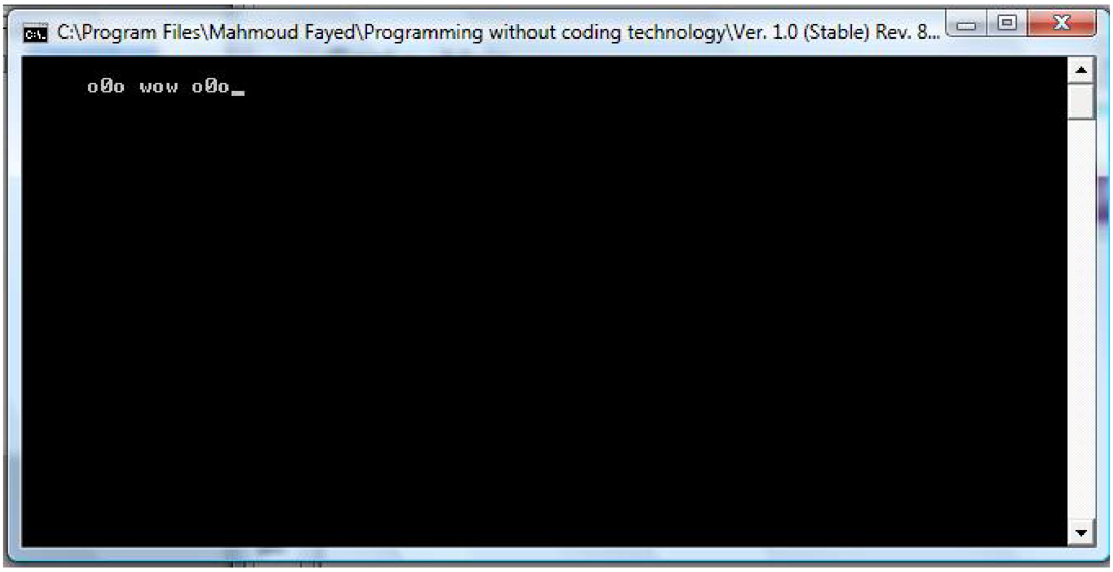
Domain (Variables) Component (Right trim)



Input variable name = output variable name = mystr



The Final Steps tree

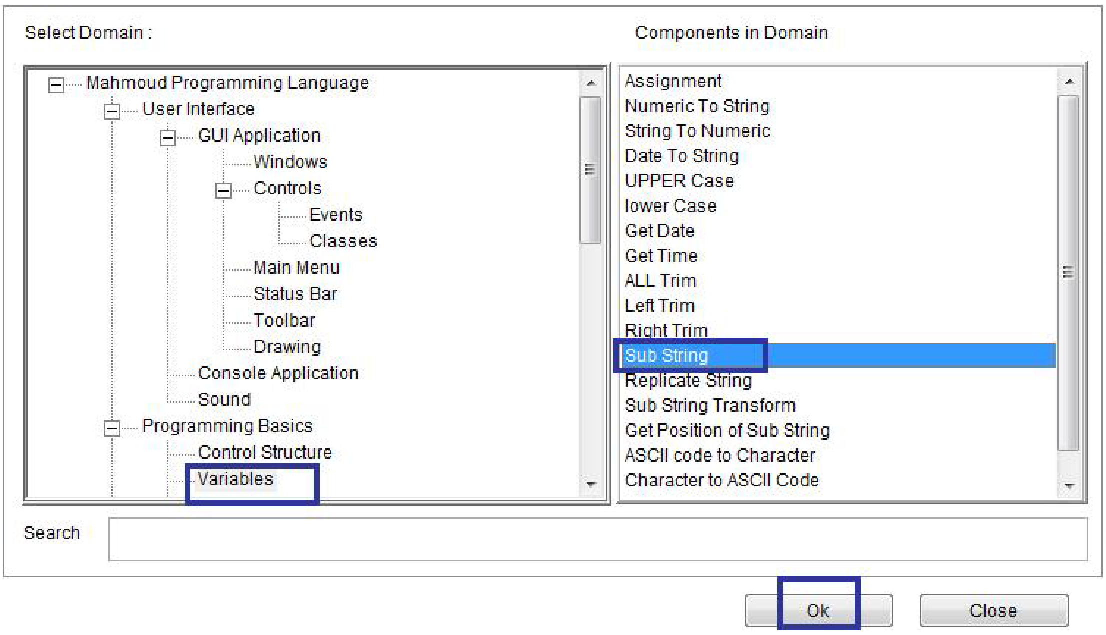


The final application

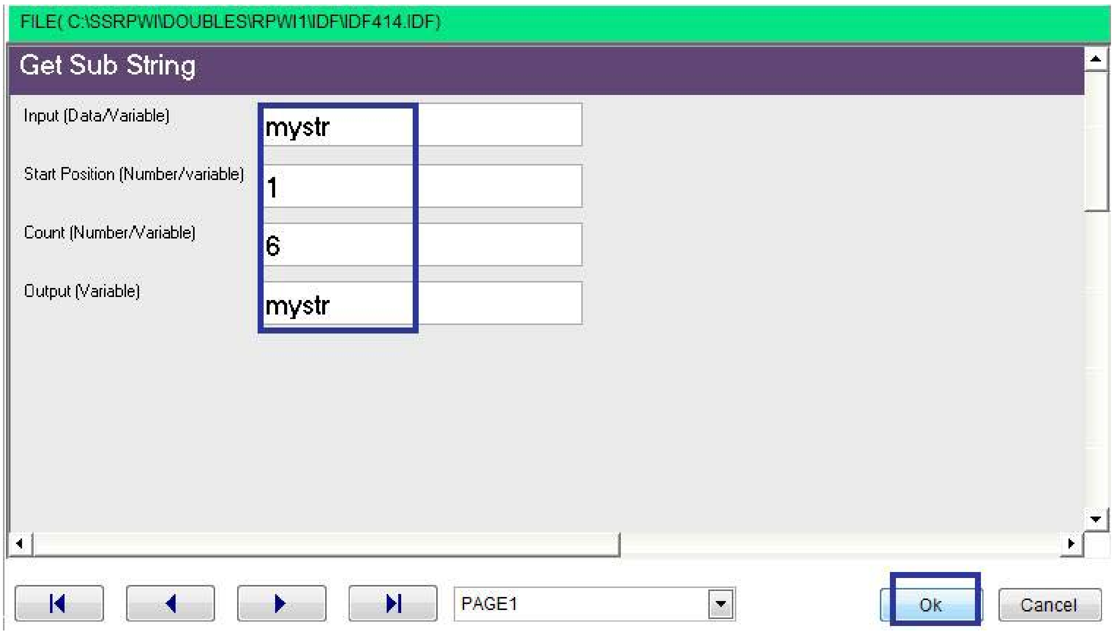
Sub String:-

Extract a substring from a character string

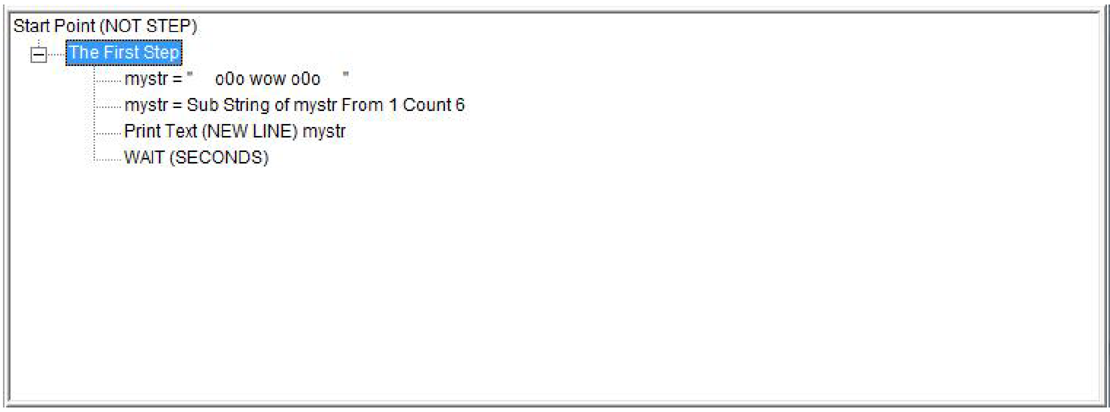
Example – Screen Shots:



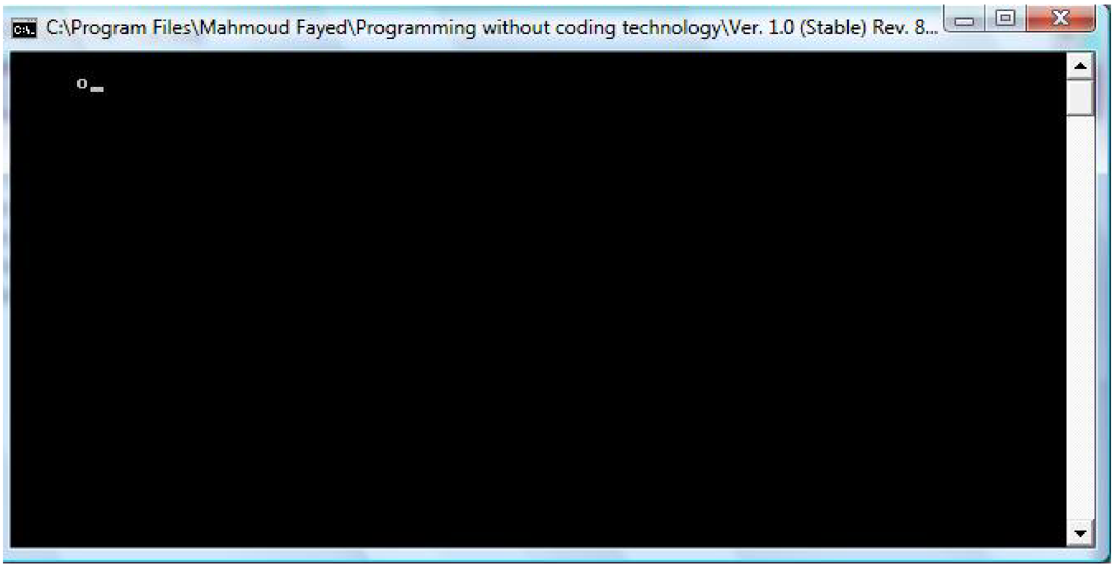
Doamin (Variables) Component (Sub String)



Interaction Page



The final steps tree

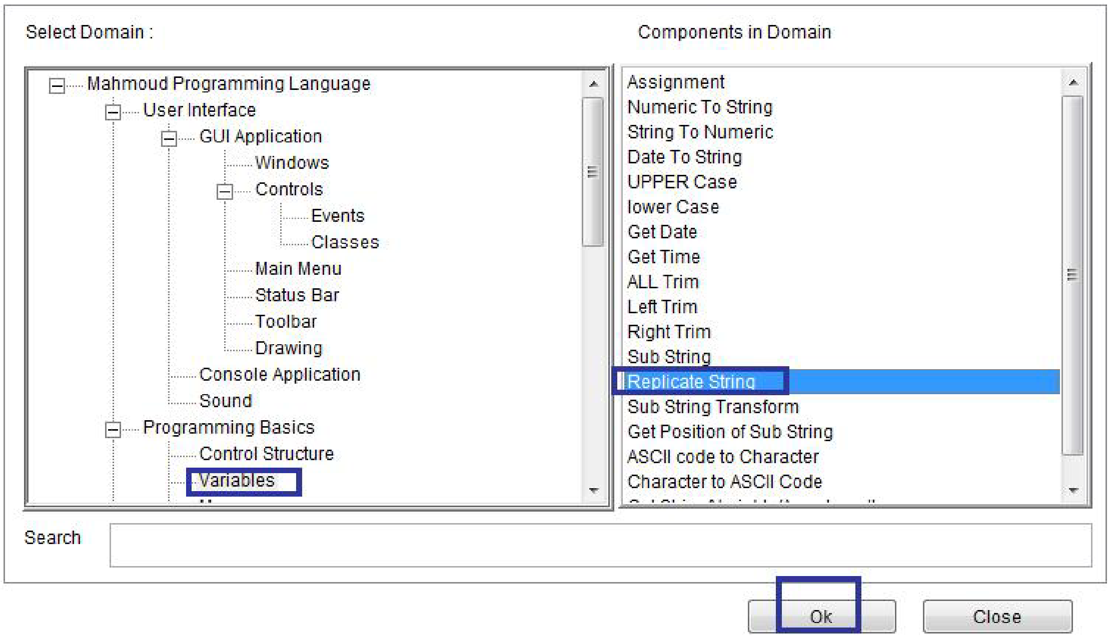


The final application

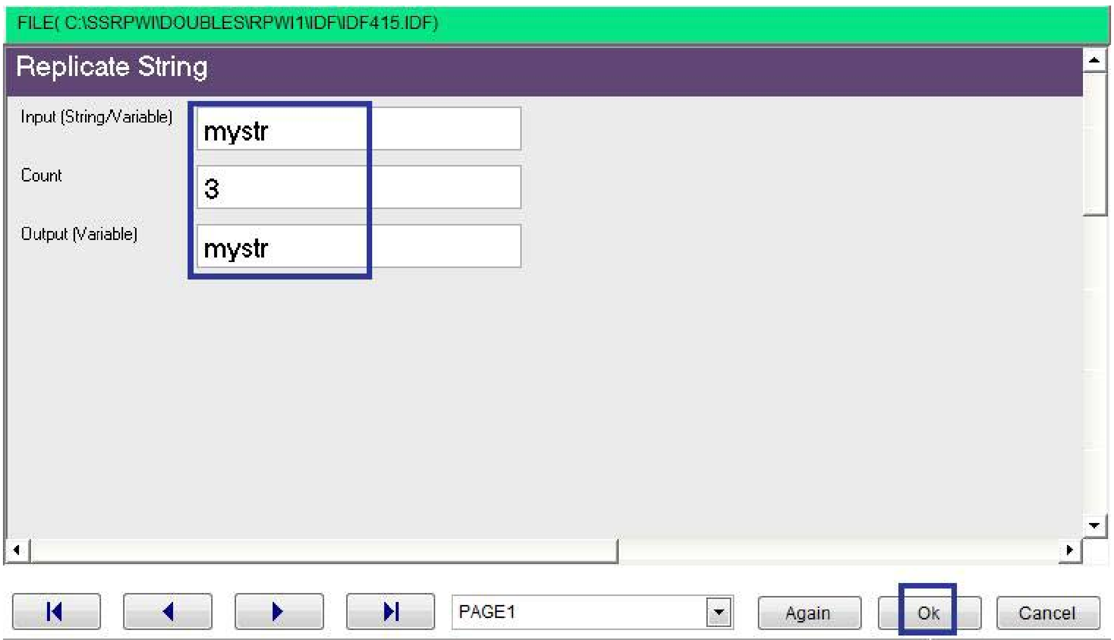
Replicate String

Returns a string repeated a specified number of times

Example – Screen Shots:



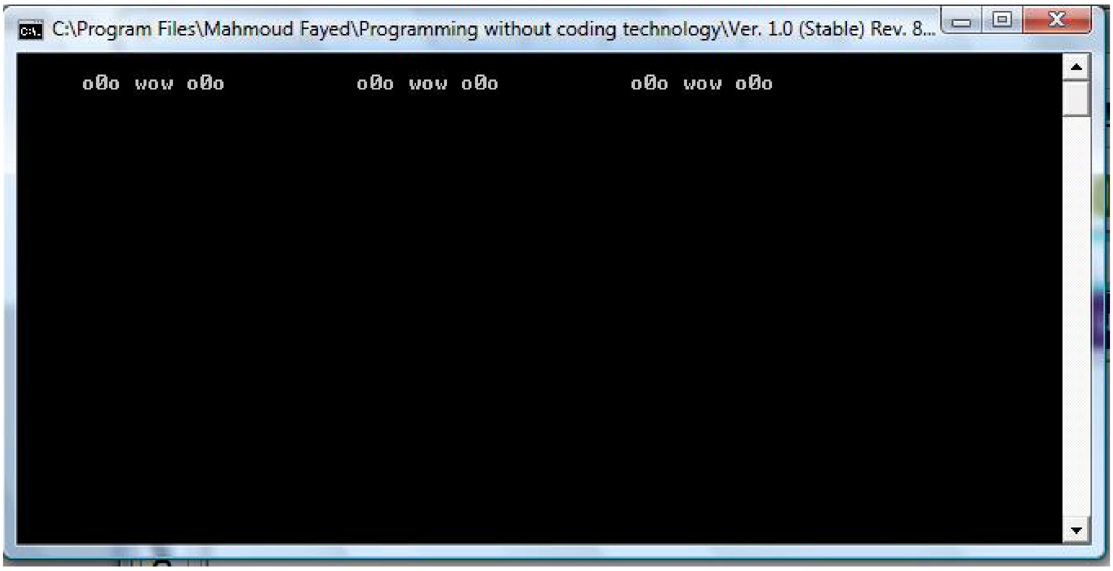
Domain (Variables) Component (Replicate String)



Interaction Page



The final steps tree

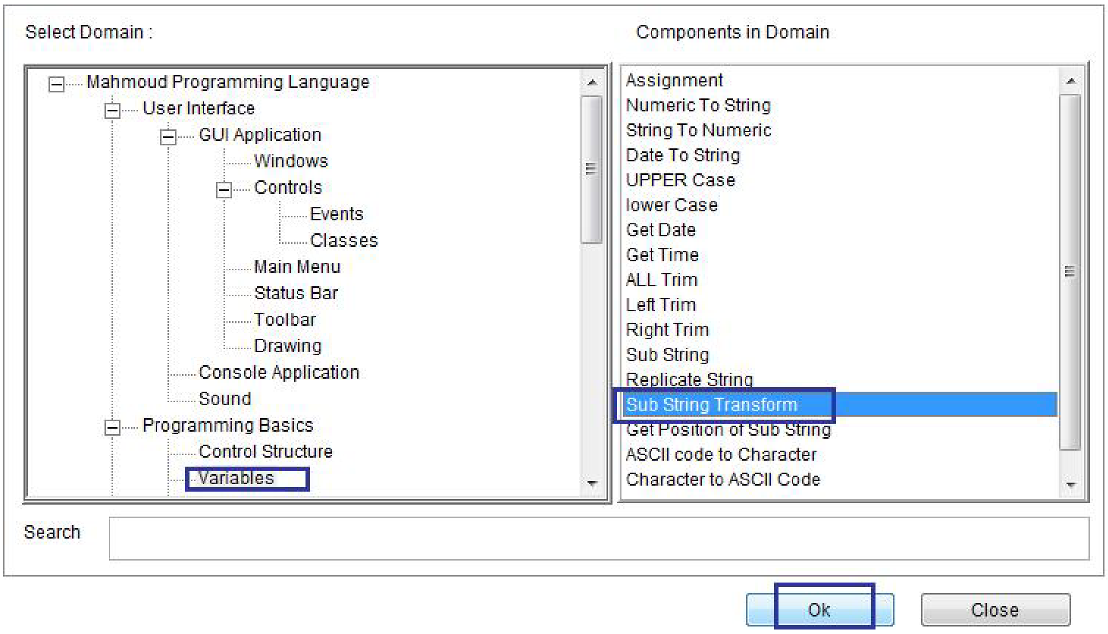


The final application

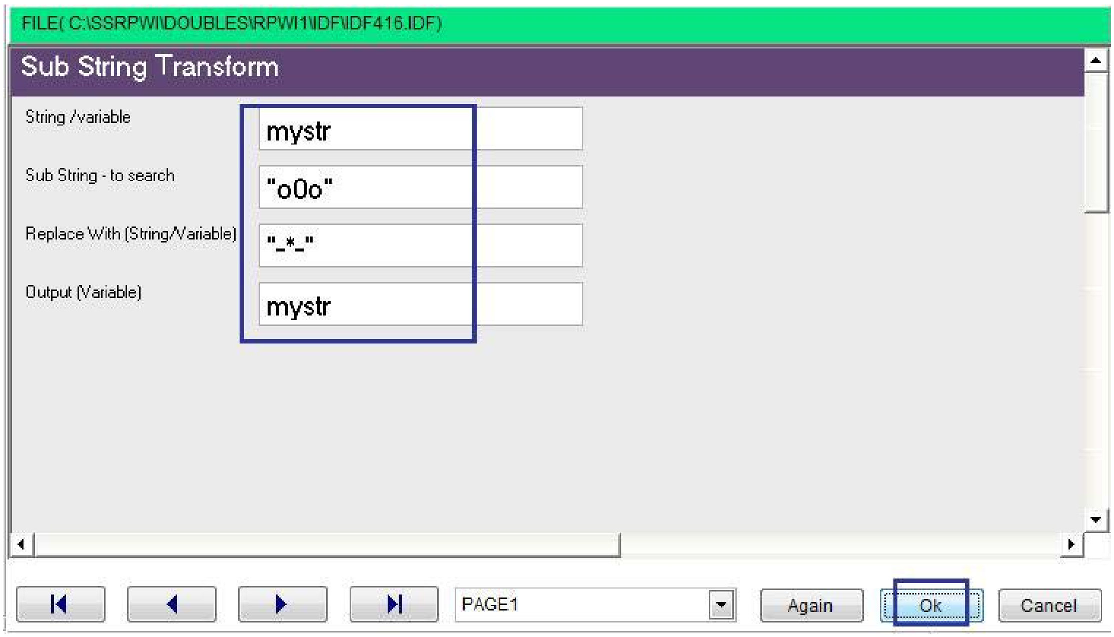
Sub String transform

Search and replace characters within a character string or memo field

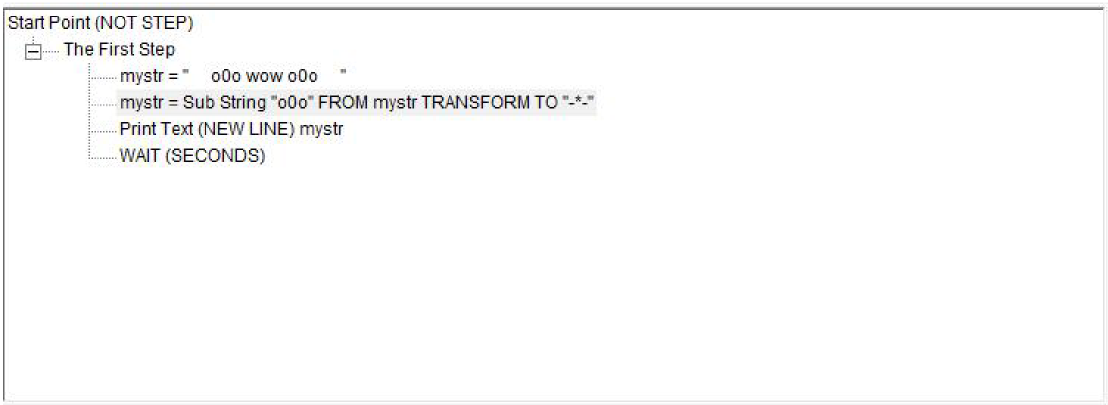
Example – Screen Shots:



Domain(Variables) – Component (Sub String Transform)



Interaction Page



Final Steps Tree